

Canvas LTI Student Climate Dashboard Mark II

Design Document

Team Number: sdmay23-42

Client: Henry Duwe

Advisers: Nicholas Fila

Team Members/Roles:

Joshua Harvey – Team Manger & Resonance/Persona/Journey Map

Colin Hasbrook – Database

Elias Simpson – Server, Kubernetes, Docker

Howard Chi - UI

Hailee Leonard – CI/CD and UX

Jonathan Giblin – Security/ Login Authentication

Team Email: sdmay23-42@iastate.edu

Team Website: <https://sdmay23-42.sd.ece.iastate.edu/>

Revised: 12/2/2022

Executive Summary

Development Standards & Practices Used

The following is the list of software practices and engineering standards used for this project. There are only software components expected for this project.

- Agile
- Acceptance / Integration / Unit Testing
- Kubernetes
- Docker
- CI/CD
- SOLID Principles
- FERPA

Summary of Requirements

Functional requirements (specification)

1. Create journey map using stored data
2. Take user input to display student, persona, and class journeys
3. Take user input to customize journey map
4. Journey map should use relative weighting based on achievement, sentiment, engagement
5. Journey map uses a visual scale and x-axis time frame are customizable
6. Journey map uses a sliding timescale for time window of data
7. Journey map uses a sliding magnitude of drop-off of older data
8. Journey map has the option to amplify data based on trend
9. Journey map can predict future trend with straight line
10. Journey Map automatically identifies trends in resonance data and displays them in shaded regions
11. Resonance can be flexibly defined in app
12. Resonance definitions can be saved and selected
13. Personas can be flexibly defined in app
14. Persona definitions can be saved and selected
15. Automation of persona definitions
16. Gather data directly from Canvas API
17. Categorize personas based on user input (proximity, student defined persona, best performing category, sentiment, engagement)
18. Present unique data points to analyze effect, trend, contrast with personas, etc.
19. Grant appropriate permissions for student to access personal journey map
20. Allow students to provide comments and feedback on specific data points which are visible to professor

Resource Requirements

1. Cloud, Kubernetes

2. API interactions
3. Server
4. Canvas
5. Docker
6. CI/CD (Gitlab)

Physical Requirements

1. Server

Aesthetic Requirements

1. Appealing and attractive layout, colors, and charting
2. Visualization choices represent meaningful data and/or user preferences
3. Intuitive design

User Experiential Requirements

1. Canvas integration
2. Easy to use
3. Adjusts to specific needs
4. Page should load within five seconds

Economic/market Requirements

1. Our project will not be designed for sale or distribution outside of ISU
2. There are currently no economic requirements for our project. All costs are subject to ISU financing, currently no technologies will require payments or subscriptions.

UI Requirements

1. For use on a full screen browser
2. Fullscreen option for journey map
3. Accommodates multiple screen sizes
4. Interactive, allowing toggling and resizing
5. *Professor*
 - a. Display resonance customization features
 - b. Display mapping customization features
 - c. Display persona customization features
 - d. Save resonance definition
 - e. Save persona definition
 - f. Load resonance definition
 - g. Load persona definition
 - h. Chart color coded journey map based on persona groups and/or students
 - i. Clicking on data points provides granular data information
 - j. Change time chunks in Journey Map
 - k. Select trend visualization highlight
6. *Student*

- a. Display resonance over time
- b. Compare map with persona groups and/or overall class

Applicable Courses from Iowa State University Curriculum

- COM S 227 – Object-oriented Programming
- COM S 228 – Introduction to Data Structures
- COM S 309 – Software Development Practices
- COM S 311 – Introduction to the Design and Analysis of Algorithms
- COM S 363 – Introduction to Database Management Systems
- COM S 362 – Object-Oriented Analysis and Design
- SE 317 – Introduction to Software Testing
- SE 319 – Construction of User Interfaces
- SE 329 – Software Project Management
- SE 339 – Software Architecture and Design

New Skills/Knowledge acquired that was not taught in courses

- Docker
- Kubernetes
- Canvas API
- Istio
- Helm
- FERPA
- Python

Contents

Canvas LTI Student Climate Dashboard Mark II	0
Executive Summary	1
Development Standards & Practices Used	1
Summary of Requirements	1
Applicable Courses from Iowa State University Curriculum	3
New Skills/Knowledge acquired that was not taught in courses.....	3
List of Figures.....	6
1 Team	7
1.1 Team Members.....	7
1.2 Required Skill Sets for Your Project	7
1.3 Skill Sets covered by the Team	7
1.4 Project Management Style Adopted by the team	7
1.5 Initial Project Management Roles.....	7
2. Introduction.....	7
2.1 Terms and Definitions.....	7
2.2 Problem Statement.....	8
2.3 Intended Users and Uses	8
2.4 Requirements & Constraints.....	9
2.5 Engineering Standards	10
3 Project Plan.....	11
3.1 Project Management/Tracking Procedures.....	11
3.2 Task Decomposition.....	11
3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria	13
3.4 Project Timeline/Schedule	14
3.5 Risks and Risk Management/Mitigation	14
3.6 Personnel Effort Requirements	16
3.7 Other Resource Requirements	17
4 Design	18
4.1 Design Context.....	18
4.1.1 Broader Context	18
4.1.2 Prior Work/Solutions.....	19

4.1.3 Technical Complexity	21
4.2 Design Exploration	21
4.2.1 Design Decisions.....	21
4.2.2 Ideation	22
4.2.3 Decision-Making and Trade-Off	23
4.3 Proposed Design	23
4.3.1 Overview	23
4.3.2 Detailed Design and Visual(s).....	25
4.3.3 Functionality.....	30
4.3.4 Areas of Concern and Development	31
4.4 Technology Considerations.....	32
4.5 Design Analysis.....	32
5 Testing	33
5.1 Unit Testing.....	33
5.2 Interface Testing	33
5.3 Integration Testing.....	34
5.4 System Testing	34
5.5 Regression Testing	35
5.6 Acceptance Testing.....	35
5.7 Security Testing.....	35
5.8 Results.....	35
6 Implementation.....	35
7 Professional Responsibility.....	36
7.1 Areas of Responsibility.....	36
7.2 Project Specific Professional Responsibility Areas.....	37
7.3 Most Applicable Professional Responsibility Area.....	39
8 Closing Material.....	39
8.1 Discussion	39
8.2 Conclusion.....	39
8.3 References	40
8.4 Appendices.....	41
8.4.1 Team Contract.....	41

List of Figures

Figure 1: Project schedule	14
Figure 2: Journey map tool from Rains Media.	20
Figure 3: Journey map from sddc21-19.....	22
Figure 4: Overview of system implementation.	24
Figure 5: Mark I's fixed definition of Resonance.....	26
Figure 6: Mark II supports flexible definitions of Resonance.....	26
Figure 7: Fade Dynamic modifier.	27
Figure 8: Visualization of identifying trends of students between persona groups.	28
Figure 9: Bell curve example, used to select students of similar resonance movement.	29
Figure 10: Mark I vs Mark 2 Persona implementation.....	29
Figure 11: UI collage.....	30

1 Team

1.1 Team Members

- Joshua Harvey
- Colin Hasbrook
- Elias Simpson
- Howard Chi
- Hailee Leonard
- Jonathan Giblin

1.2 Required Skill Sets for Your Project

- Kubernetes, Docker, API, JavaScript, Python, SQL, CI/CD, Database, Security

1.3 Skill Sets covered by the Team

- Kubernetes: Elias, Jon
- Docker: Elias, Colin, Jon, Hailee
- API: Elias, Jon, Josh
- JavaScript: Hailee
- Python: Howard, Colin
- SQL: Hailee, Colin
- CI/CD: Hailee
- Database: Colin
- Security: Jon

1.4 Project Management Style Adopted by the team

- Agile – We will employ agile values and principles. This will help respond to user needs, prioritize tasks, and improve the quality of work. It increases the probability of project success by providing an adaptive workflow.

1.5 Initial Project Management Roles

- Joshua Harvey – Team Manager & Resonance, Persona, Journey Map
- Colin Hasbrook – Database
- Elias Simpson – Server, Kubernetes, Docker
- Howard Chi – UI
- Hailee Leonard – CI/CD, UX
- Jonathan Giblin – Security, Authorization

2. Introduction

2.1 Terms and Definitions

- Persona – Students are sorted into groups based on various metrics; these groups are then graphed accordingly. “Persona” refers to a specific grouping of students. Default grouping is based according to students with similar scores across distinct categories.

Separating the entire class of students into persona groups helps narrow down the effects that course events have on students.

- Journey Map – The journey map is a visual representation of the resonance of a student or a group over time. The X axis represents time; the Y axis is the resonance for at that point in the course. By visually representing the data, the instructor and student can gain new insights that can drive class design and personal behavioral change.
- Resonance - A loose term for tracking student engagement with the course throughout the semester. Because the app aims to offer full access to its algorithmic process through APIs, the definition of what resonance means is purely a function of the algorithm's current configuration. For example, the model of resonance based on achievement, sentiment, and engagement aims to track student resonance through an understanding of a student's scores across the three metrics. The inputs for the model might be customized or reconfigured to investigate resonance on a more granular level, or to look for subtle correlations between inputs.

2.2 Problem Statement

The problem we are addressing is the lack of instructor and student resources that analyze student and course data to predict student resonance with the course. Instructors need a way to accurately understand students' experiences in their courses, so they can empathize with students, improve student interactions, and course design. Students need a tool that allows them to reflect on their class performance and use it as a basis for instructor communication. Currently, ISU's course management software, Canvas does not have a solution for in-depth automated data analysis that measures students' resonance with the course.

Over the course of a semester, instructors and students are missing out on feedback that could dynamically influence student resonance with the course. The problem we are addressing is preventing instructors from accurately understanding their students' experiences with the course and how each could change their respective actions to improve student resonance and course achievement. Our solution uses automatically generated Journey Maps to visualize student and course data in an interactive application that enables instructors and students to discover new insights that can be used to change course design and student interactions with the course material.

2.3 Intended Users and Uses

We divided users into two groups, faculty, and students. For each group we identified personas that elaborated on specific needs the users would require and how they would benefit from the software tool. The faculty is made up of three different personas that have different needs and goals for the application. Tim is educated on the tool and interested in using it to improve his course and student experiences. Bones is more of an opposite to Tim; he is not educated on the tool and is only using it because it is a department requirement. Doug is a department chair and is interested in results from classes that are using the tool, so he can best make changes to improve the courses in the department. We have three different student personas that incorporate the most common needs and goals from using the tool. Summer is the ideal student and eager to get the most out of the tool for her classes. Chad is not the ideal student and will

not engage with the tool except when required. Colby is a graduate student and will be using the tool as a TA. He will be using the tool to better understand students and report trends to the professor and offer advice when possible. Our selection of personas explores the most common and important users and uses of the system. Using these personas when determining requirements of the systems will help us ensure most users have benefited from consideration of their top needs, helping them get the most out of the application.

2.4 Requirements & Constraints

Functional requirements (specification)

1. Create journey map using stored data
2. Take user input to display student, persona, and class journeys
3. Take user input to customize journey map
4. Journey map should use relative weighting based on achievement, sentiment, engagement
5. Journey map uses a visual scale and x-axis time frame are customizable
6. Journey map uses a sliding timescale for time window of data
7. Journey map uses a sliding magnitude of drop-off of older data
8. Journey map has the option to amplify data based on trend
9. Journey map can predict future trend with straight line
10. Journey Map automatically identifies trends in resonance data and displays them in shaded regions
11. Resonance can be flexibly defined in app
12. Resonance definitions can be saved and selected
13. Personas can be flexibly defined in app
14. Persona definitions can be saved and selected
15. Automation of persona definitions
16. Gather data directly from Canvas API
17. Categorize personas based on user input (proximity, student defined persona, best performing category, sentiment, engagement)
18. Present unique data points to analyze effect, trend, contrast with personas, etc.
19. Grant appropriate permissions for student to access personal journey map
20. Allow students to provide comments and feedback on specific data points which are visible to professor

Resource Requirements

1. Cloud, Kubernetes
2. API interactions
3. Server
4. Canvas
5. Docker
6. CI/CD (Gitlab)

Physical Requirements

1. Server

Aesthetic Requirements

1. Appealing and attractive layout, colors, and charting
2. Visualization choices represent meaningful data and/or user preferences
3. Intuitive design

User Experiential Requirements

1. Canvas integration
2. Easy to use
3. Adjusts to specific needs
4. Page should load within five seconds

Economic/market Requirements

1. Our project will not be designed for sale or distribution outside of ISU
2. There are currently no economic requirements for our project. All costs are subject to ISU financing, currently no technologies will require payments or subscriptions.

UI Requirements

1. For use on a full screen browser
2. Fullscreen option for journey map
3. Accommodates multiple screen sizes
4. Interactive, allowing toggling and resizing
5. *Professor*
 - c. Display resonance customization features
 - d. Display mapping customization features
 - e. Display persona customization features
 - f. Save resonance definition
 - g. Save persona definition
 - h. Load resonance definition
 - i. Load persona definition
 - j. Chart color coded journey map based on persona groups and/or students
 - k. Clicking on data points provides granular data information
 - l. Change time chunks in Journey Map
 - m. Select trend visualization highlight
6. *Student*
 - n. Display resonance over time
 - o. Compare map with persona groups and/or overall class

2.5 Engineering Standards

Below is our list of Engineering standards that we will use during the development process and in the developed application. We identified software development standards learned from previous classes and are common solutions in industry. Following these standards will help steer our efforts and ensure we are developing a safe and secure application.

Agile – We will employ agile values and principles. This will help respond to user needs, prioritize tasks, and improve the quality of work. It increases the probability of project success by providing an adaptive workflow.

Acceptance / Integration testing – We will continually ensure that our software is meeting customer requirements and integrating with the system.

Kubernetes – Kubernetes is an open-source system for automating deployment, which affects the scope and management of “containerized” applications. This is a great system for any project to reduce unnecessary spending and to apply changes to our project. Which is why we will continue to use Kubernetes for our project

Docker – Docker is a platform that is used for developing, shipping, and running applications. This is beneficial to provide applications with different services for this project, especially being able to implement continuous integration of the application.

CI/CD - We will use GitLab for CI/CD software development. Pull requests will require team approval and will automatically be tested on creation. Strong checks for updates to the environments will reinforce accountability and maintainability.

SOLID Principles – Solid software design principles will allow for greater flexibility for later additions and modifications.

FERPA – follow all FERPA rules and guidelines when handling student data and information.

Testing – will follow the agile development process and implement various testing activities.

3 Project Plan

3.1 Project Management/Tracking Procedures

We will be implementing an agile project management style. Agile will give us the flexibility to build out the application in an adaptive manner, allowing regular client feedback and less definitive planning upfront. We anticipate challenges stemming from learning and implementing new technologies, integrating a long list of functionalities and flexibility, and incorporating the application into a self-contained application. Agile will give us the flexibility to make progress, adapt to successes and failures, and improve the odds of project success.

We will be using Jira for agile issue, sprint, and project management. Jira allows for issues to be assigned to group members, providing clear task responsibilities and progress. Each sprint will include additional project management tools and metrics. GitLab will be used as our code repository and solution for CI/CD development. Figma will be used for UI sketches and ideas that can be shared within the team and with the client for feedback. Additional tools such as Discord and Microsoft Teams will be used to manage team functions and communication.

3.2 Task Decomposition

UI:

Project	Priority	Description
---------	----------	-------------

New customization features 1. Time frame journey map 2. Fade dynamic journey map 3. Trend enhancer journey map 4. Persona selection 5. Persona definition 6. Resonance selection 7. Resonance definition 8. User code input for resonance definition	High	Ability to enter desired settings for customization options into user interface
Student interface	High	Ability for student users to view their personal journey map—and that of persona groups/overall course
Full screen capability	Medium	Fullscreen button which enables maximization of journey map
Redesign for space and time efficiency	Low	Tables and input options take up only as much space as needed, leaving more room for the map

API:

Project	Priority	Description
Customized settings get to logic	High	Customization properties passed to the algorithm
User type gets to logic	High	User type is transported to logic layer through response properties

Logic:

Project	Priority	Description
Flexible definition of resonance	High	Take the changes from the API and incorporate those changes into the possible combinations of metrics used to score resonance
Flexible definition of personas	High	Flexible definition of personas will allow for heightened specification of student types and desired understanding by instructor
Automatic generation of personas	High	Personas are automatically generated based on the

		resonance components scores
Scripting definition of resonance	Medium	Alternative means of defining resonance outside of the user interface controls
Resonance and Persona Accuracy Testing	High	Send the correct data to be viewed by the different users (instructors all student data and students just their own)

Integration:

Project	Priority	Description
Simple setup for courses	Medium	Available for multiple courses with intuitive setup
Accessing FERPA data for development	Medium-high	Ability to use full-course data without breaching privacy laws (anonymization)
Okta verification for user authentication	Medium	Ability for instructors and students to access the application using their ISU accounts

3.3 Project Proposed Milestones, Metrics, and Evaluation Criteria

Milestone	Measurement
Setup existing infrastructure, Kubernetes, Docker, Okta, etc. (Infrastructure)	The past team's application loads and is functional.
Full screen journey map view (UI)	User can expand journey map for intimate analysis
Student has a unique view and limited functionality (UI)	User can interact with their personal data and anonymized class data
User interface organizes controls and allows for faster changes in analysis (UI)	Controls are customizable in position and easy to access when analyzing changes in journey map parameters
Journey map segment length (UI)	Journey map segment lengths can be modified and include various components of the resonance definition
Add python file for defining resonance and persona definitions (Logic)	Users can upload files that correctly set the definitions used for resonance and persona classification
User can manually select the impact past events have on current resonance with the course (Logic)	Output changes to better reflect the time span of one event and its impact on others

User Interface can automatically highlight sections of student or class trends (Logic)	Users can intuitively see trends when highlighted on the journey map
All API communications will get or send a response within three seconds (API)	The test accurately measures the response time for most common pulls from Canvas
The application is integrated with developed instance of Canvas (Integration)	Canvas API features are functional with the application
Application can seamlessly switch from our Okta instance to ISU's instance (Integration)	Documented changes allow for easy setup of the application for real courses

3.4 Project Timeline/Schedule

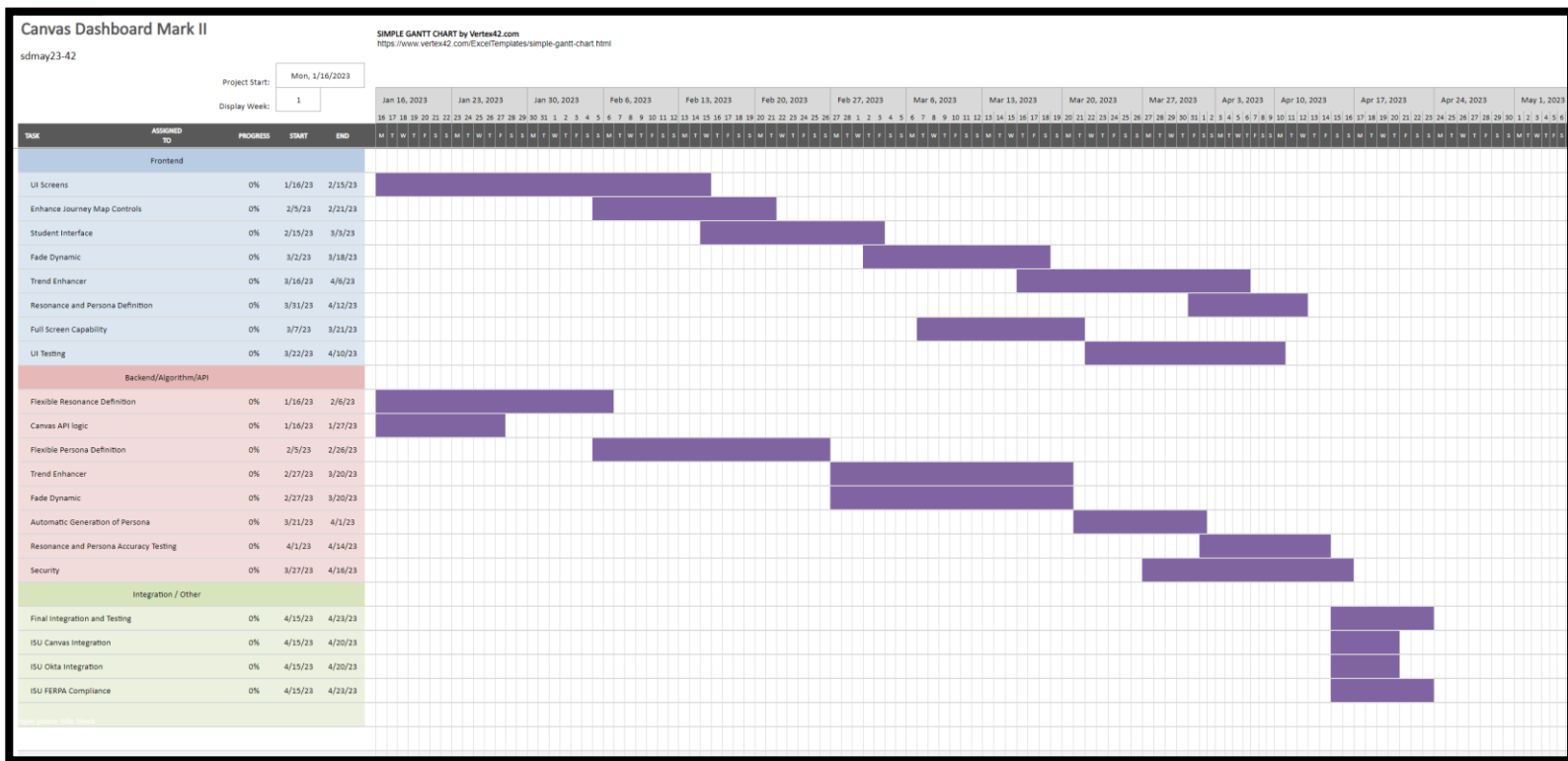


Figure 1: Project schedule

3.5 Risks and Risk Management/Mitigation

Task	Risk	Mitigation	Estimate Probability
------	------	------------	----------------------

Infrastructure	Virtual machine becomes inaccessible.	Setup a cluster so the service can be transferred to alternate sessions.	.1
Infrastructure	Virtual machine no longer provided by ETG.	Find a cloud provider.	.05
Data Storage	Data is deleted from the database.	Keep backup on GitLab. Preventative measures so data can't be mistakenly or maliciously deleted.	.1
Data Storage	Data is compromised.	Follow FERPA and IEEE standards. Ensure data is sufficiently secured and plan is made to handle situation if occurs.	.2
Data Storage	Student data can't be effectively secured.	Additional assistance may be required if we have difficulty meeting FERPA and IEEE standards.	.1
Technical Ability	Learning past teams code and building from it.	The past teams code may require additional investment to understand fully. If their code can't be used for some of our new features, additional time may be required to implement alternatives.	.8
Canvas Integration	Canvas API change, leading to outdated requests.	Manual modification of code will be required.	.1
Canvas Integration	Can't refresh data from Canvas in a timely manner.	Experiment with different caching or volume of requests.	.2
Canvas Integration	Application can be integrated in Canvas application.	Will only be able to post link in Canvas.	.3
Algorithms	Functions of new algorithms involved in calculating resonance, personas, and journey map are not known until later in development.	Schedule must be flexible when implementing new algorithms.	.7

Algorithms	Portrayal of students is not accurate and leads to destructive actions by the instructor.	The tool will not be used like Mark I	.5
Team	Intended features are not able to be implemented in Mark II	Mark III would need to be planned and include quality documentation.	.8
Team	Loss of team member or leave due to illness.	Reference dynamic tasks and goals document, and our three prototypes, select what will need to be dropped from Mark II.	.1
Team	Team is making slow progress.	Investigation of the causes will take place. Interventions will be used if the problems found are effort or behavior related. Technical issues will be subject to additional investigation or release from Mark II.	.4

3.6 Personnel Effort Requirements

Task	Education, setup, and integration (hours)	Implementation (hours)	Explanation
Time Frame for UI	2	4	HTML slider, UX.
Fade Dynamic for UI	2	4	Two HTML sliders, one for drop off effect, one for linear/exponential drop off, UX.
Trend Enhancer for UI	2	4	HTML toggle switch, UX.
Persona Groups for UI	2	4	HTML dropdown for number of groups, UX.
Resonance Definition	3	5	Selection of resonance definition, toggle of parameters.
Student Interface	5	20	Will rely on existing logic and API but requires incorporating user type to logic, refactoring or code, and implementation of new user interface.
Full Screen Capability	2-10	5-20	Expanding a designated portion of the application could require extensive

			modifications to the modules and/or framework of the app.
Space & Time Efficiency	2	4	HTML/CSS layout.
API modifications for new customization features and user types	3	6	Details about classes, serializers, and data flow will need to be investigated.
Canvas Integration	5-20	10	Most of the set-up time will go towards learning how to work with Canvas environment.
Algorithm: time frame	5	15	Python and UI working closely to setup time frames and correct scoring.
Algorithm: linear/exponential drop-off	5	15	Python and UI working closely to tune resonance based on time frame.
Algorithm: drop-off strength	5	15	Python and UI working closely to tune resonance based on time frame.
Algorithm: trend enhancer	5	15	Python and UI for visualizing the automatic notification of identified trends in resonance and time periods.
Persona Definition	10	20	Python for creating persona scores based on resonance component scores.
Resonance Definition	10	20	Python for creating resonance scores based on individual component scores.
Persona grouping: number of groups	4	1-3	Most of this will go into planning, adding leeway time since things never go according to plan.
Persona grouping: grouping based on input factors	4	1-3	Mainly just looking at data and deciding what goes where.
ISU FERPA Compliance	6	15	Begin setting application up for potential ISU FERPA certification.
ISU Okta Integration	2	2	Setup to integrate with potential ISU integration.

3.7 Other Resource Requirements

There should be no parts or materials that are required to complete the project. The software will run on existing University equipment and there are no planned components that will require additional money or equipment.

4 Design

4.1 Design Context

4.1.1 Broader Context

Public health, safety, and welfare:

This project has no significance to public health and safety since it is an application to see how different students are doing and feeling about the current course that they are enrolled in. As for welfare it is important for the resonance of the application to give an accurate result for the students and teacher to understand how they are doing and feeling about a course. This should increase their feelings towards the class and help them see more clearly where they are struggling. From the instructor's perspective this would be good for their own personal welfare, to understand their students' needs and where the course can be improved to accomplish this.

Global, cultural, and social:

University: Overall, the journey map dashboard strives to improve student experience and academic culture at Iowa State University. It helps to bridge the gap between professors and students by offering a comprehensive reading of not just student performance, but also sentiment and participation.

Administration: Administrative culture seeks to improve communication and trust between all members of the university. The journey map provides a powerful tool to study and improve the academic environment.

Professors: It offers a valuable tool for professors seeking to improve courses and better understand the student experience. It's a way to improve engagement with students and monitor student responses and performance across time. The ability to customize the map according to desired metrics allows professors to hypothesize and experiment with the tool, attempting to identify and/or isolate student groups who most resonate with the course--and those most in need of changes/different approaches. Finally, it can be used to address individual student needs and understand how students relate to other students within the course.

Students: Student culture benefits both directly and indirectly from the application. Students can use the dashboard to view their resonance with a course across time, offering a valuable ability to externalize and add depth to their perspective of a course. It allows them to look for trends, patterns, or details that help them assess and evaluate their performance, engagement, and sentiment over time, helping them set and achieve academic goals. Indirectly, student culture benefits from the trust that each student's personal experience will be considered by the professor in future iterations of the course, thus offering up their "data" as a valuable tool which could improve the experiences of future students with similar academic traits.

Environmental:

This project is made of software for academic purposes, so it has little relation to the environment. Direct environmental impact will be non-existent as a result since it is just software. Indirect environmental impact is expected to be minimal as the only indirect impact

our project will have on the environment is energy usage for servers to run the software. Since our software is not expected to be energy intensive this indirect impact should be low.

Economic:

The current scope of the project has limited economic implications for ISU and its users. The project was developed from initial research funded by NSF and completed by ISU faculty. ISU faculty have continued to develop the concepts learned from the prior research through Senior Design Projects. All future costs are subject to ISU allocations for Senior Design Projects, generally available student resources, and potential funds secured by the project's client. The tool we are developing could have many possible economic benefits and costs associated with its implementation. The tool could improve instructor's understanding of students and course, allowing the instructor to make better use of resources and benefiting the students' educations. The university could benefit from instructors using the information gathered from the tool to efficiently improve classes and student experiences. Students could be more engaged in classes and rely less on the traditional reactionary student resources available in most courses. The tool could also negatively impact instructor and student interactions and cause a waste of resources or worsening class climate. The tool currently has no intention of being used by other universities or sold as a product. The economic context of our tool is solely focused on a usable product, with no or very few ISU funds, for the Electrical and Computer Engineering Department at ISU.

4.1.2 Prior Work/Solutions

Journey maps are traditionally used in a business context to aggregate customer experiences and identify aspects of customer service that can be improved. Their application is generally highly conceptual and theoretical, a prompt which helps managers plan and map out a customer experience. Some journey maps based on specific user responses and data have been applied to a university setting. However, existing commercial products focus on a more general student

experience—such as onboarding—and less with a specific course. For example, Rains Media offers a product to help analyze the recruitment of prospective students visualized in figure 2.

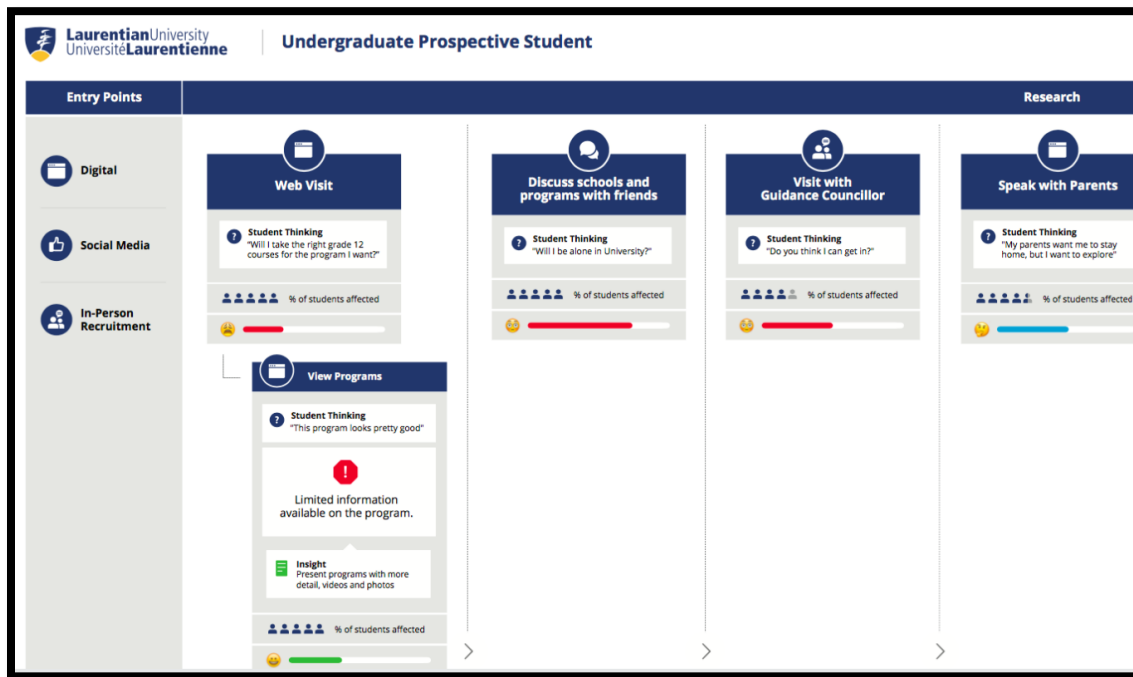


Figure 2: Screen shot of the journey map tool on Rains Media website.

The only known attempts at applying data-based journey maps to a course setting are part of a grant from the National Science Foundation. The grant enabled professors and teaching assistants the time and resources to plan and implement journey maps for engineering courses. The drawbacks of these early attempts at journey maps were that they involved manual accumulation, recording, and mapping of data. The study does indicate that journey maps improve professor empathy with students and leads to greater understanding of student issues and performance.

We thus have the benefit and disadvantage of being pioneers in the space. In general, an academic course offers a high potential reward for the application of a journey map, due to the abundance of data provided throughout a semester.

For our project we will be following the work of a previous Senior Design group that worked on the first iteration of this LTI Canvas Resonance tool. For their advantage it has a working interface for the instructor. The journey map lays out the assignments and lets the instructor change some features. This iteration did have some shortcomings, which is our goal to improve in our iteration. The first of these was the issue with the resonance algorithm. As it stands it does not do a great job giving the correct resonance information and needs to be improved in that regard. Second would be the issue that it currently cannot connect directly within the Canvas API. The work around that they used was to use their own Okta to run the application, but this would not serve the benefit of students as is the goal of the design. The final shortcoming is it does not have a student interface, so nothing is provided to the students.

Pros – Accurate resonance algorithm, allows instructors to define resonance and personas flexibly, student facing interface.

Cons - possible issues with Canvas implementation, Okta verification, ISU integration.

4.1.3 Technical Complexity

The design utilizes containerized components as microservices. The design implements a Kubernetes cluster for container management. Storage is handled by the MySQL database. The technologies and programming languages used are strong and effective tools for meeting our requirements. Our project also incorporates journey maps, personas, and resonance with proprietary algorithms that are being applied in novel ways.

Our tool is being designed to be one of the first applications of journey maps and personas in Engineering course design. It is a challenge to apply standardized concepts to new applications. We will be continuing to define scope and requirements as we understand what is possible with the new tool. Challenges include journey map plotting, persona creation, algorithm accuracy, and features that make the tool a compelling solution for course design and instructor student understanding will be paramount in our success.

4.2 Design Exploration

4.2.1 Design Decisions

1. The UI layout for the instructor interface needs to be redesigned for easier usability and understanding.
 - a. The past implementation did not provide an easily usable interface that produced the results needed, thus making the tool unusable as a whole. The users need to want to use the tool for it to be helpful and it must be easily understandable without too many instructions or learning time. Some instructors will not want to use the tool, but will only use it because it is required, thus we need to make the UI as friendly to those users as possible.
2. The student-facing interface needs to closely resemble the instructor-facing interface for better understanding for all users.
 - a. It is assumed that some users will struggle to understand how to use the tool no matter what. Having all interfaces (student and instructor) be as similar as possible allows all users to help each other and we as developers can create a more cohesive and comprehensive user guide.
3. The journey map will be interactive to allow filtering through toggles.
 - a. The journey map is one of the most important and useful parts of the tool and must be flexible to meet all users' needs. The interactions will include filtering, resizing the map (full screen), and downloading in multiple formats (pdf or image file). Having the map be interactive allows each user to decide what factors they want or need to focus on to best learn from the results and create a plan for improvement.

4. Currently, both individual students and groups can be viewed at the same time, but there are few use cases where this is a wanted functionality, so we propose having two views. One option is to have two tabs: one for viewing students and one for viewing groups. Another option for student vs group views is to offer a toggle that will automatically change the available filters and only show one or the other.
5. The current algorithm provides limited customization and a seemingly flat journey map. The ability to control algorithm results makes the application more engaging and offers more potential discoveries for the users.

4.2.3 Decision-Making and Trade-Off

The process for identifying pros and cons of each option is to look at our use cases involved with each option and user personas to decide which fits best with all users' needs. We also get continuous feedback from users and the advisor to decide if adjustments or new options need to be made. We have not made a final decision because we have not implemented the UI, yet. We will work closely with the user and our advisor to make the best decision based on their feedback.

4.3 Proposed Design

4.3.1 Overview

Our proposed design will utilize and build from the Mark I design. The program will be a local application on the user's internet browser and a distributed system. This core design worked well for that last team, and we want to keep it going and add to it with our iteration. The application connects with the Canvas API and requires user authentication with Okta before class or student data can be displayed. FERPA standards have been acknowledged and the application will be designed with the aim of certification for broader ISU use at a later date.

Taking advantage of the microservices architecture from the first iteration, we will confine our additions to make use of this design. Many of the existing components will be expanded to account for the flexibility for dynamic definition of resonance and persona models, and new features for the user interface and student facing visualization.

The following is an overview of the key components and a diagram of their implementation in the application. This then followed by additional key terms and definitions

Key Components:

Frontend – Displays persona map, takes input from user to control/modify algorithm.

Canvas API – Interface for accessing course data from Canvas.

Local Database – Stores saved template configurations for users.

API Gateway – Microservice API which coordinates between Canvas API, local database, and frontend

Engagement Scorer – Returns the engagement score, taking applicable/selected elements as input.

Achievement Scorer – Returns the achievement score, taking applicable/selected elements as input.

Sentiment Scorer – Returns the sentiment score, taking applicable/selected elements as input.

Clustering – Creates persona outputs based on grouping criteria.

Resonance Scorer – Returns the overall resonance, taking calculated categories (grouped according to personas) as inputs.

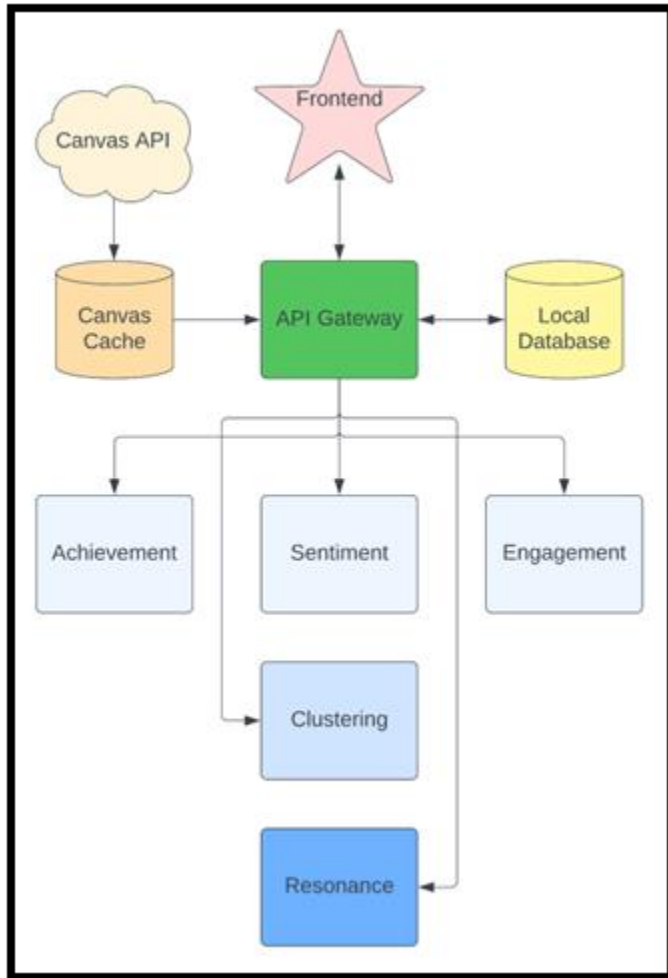


Figure 4: Overview of system implementation.

Terms and Definitions:

- Persona – Students are sorted into groups based on various metrics; these groups are then graphed accordingly. “Persona” refers to a specific grouping of students. Default grouping is based according to students with similar scores across distinct categories. Separating the entire class of students into persona groups helps narrow down the effects that course events have on students.

- Journey Map – The journey map is a visual representation of the resonance of a student or a group over time. The X axis represents time; the Y axis is the resonance for at that point in the course. By visually representing the data, the instructor and student can gain new insights that can drive class design and personal behavioral change.
- Resonance - A loose term for tracking student engagement with the course throughout the semester. Because the app aims to offer full access to its algorithmic process through APIs, the definition of what resonance means is purely a function of the algorithm's current configuration. For example, the model of resonance based on achievement, sentiment, and engagement aims to track student resonance through an understanding of a student's scores across the three metrics. The inputs for the model might be customized or reconfigured to investigate resonance on a more granular level, or to look for subtle correlations between inputs.
- Templates – A family of categories, each with customizable weights, which are used to calculate the resonance.
- Categories – A family of factors which, combined, form a template.
- Factors – The most granular element used in an algorithm to determine resonance. Elements from the course are used to form factors: grades, journal text, and attendance/turn-in rates.
- Elements – Automatically defined as specific factors based on assignment type and available data, elements provide data points which are the basis of calculations.
- Achievement, sentiment, engagement – The three categories used in a default template.

4.3.2 Detailed Design and Visual(s)

The previous team's project was a strong start and will be used as our foundation for Mark II. Mark I was thinking ahead with their application infrastructure. Using microservices as the foundation of the application will allow us to add new services with little interference with the existing design. We have not identified any limitations to the technologies, languages, and communication standards our design builds off from.

The resonance definition will be improved by allowing for flexible resonance definition and intuitive UI controls for fine tuning the output for the journey map. The previous team hardcoded definitions for Achievement, Sentiment, and Engagement, and then allowed the individual components weights to be changed in the UI. A breakdown of their definition can be seen in figure 5. By opening up the possible definitions of resonance, the user can further experiment with resonance and save definitions that visualize what they are interested in. There will be default definitions available that target each of the three main components of resonance, along with those designed for our client. Figure 6 visualizes the design change being made in Mark II.

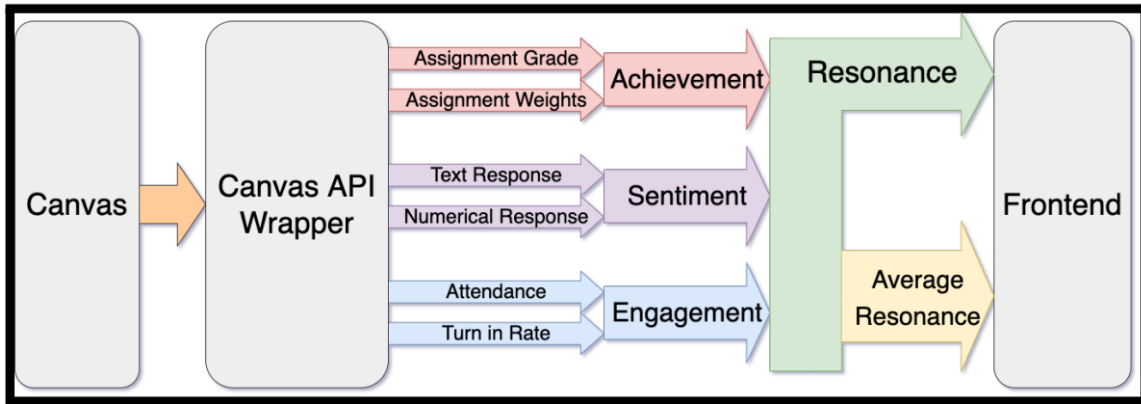


Figure 5: Mark I's fixed definition of Resonance. Taken from sddc21-19 492 Project Poster.

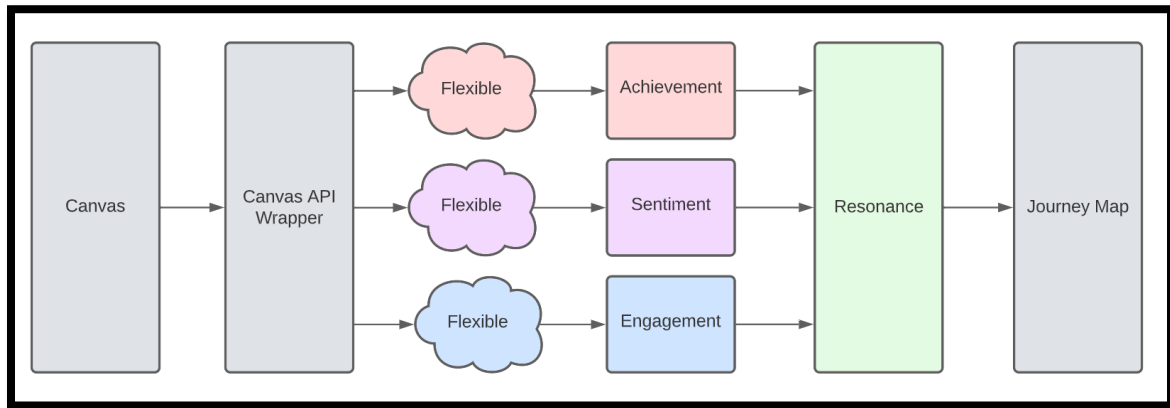


Figure 6: Mark II supports flexible definitions of Resonance using available data from Canvas.

Resonance will also need to handle one new modifier and one new functionality that further tunes the resonance scores and how they are plotted and visualized in the journey map. The modifier is called the fade dynamic, it controls the impact past, current, and future resonance input values have on resonance. This feature is designed to provide user control over how time affects scores used in the resonance definition. This feature will be driven by controls in the updated user interface. The controls apply a percentage to the resonance score's weight to amplify or dampen the value of independent scores based on their time of occurrence. This has a similar effect as changing the weights of the components used to score resonance, with the exception that this feature will only impact components that were completed during the specified time frame. The current implementation allows for UI graph manipulation by modifying the weights of assignments and achievement, sentiment, and engagement categories. We will be deploying this functionality to time frames represented on the x-axis of the journey map.

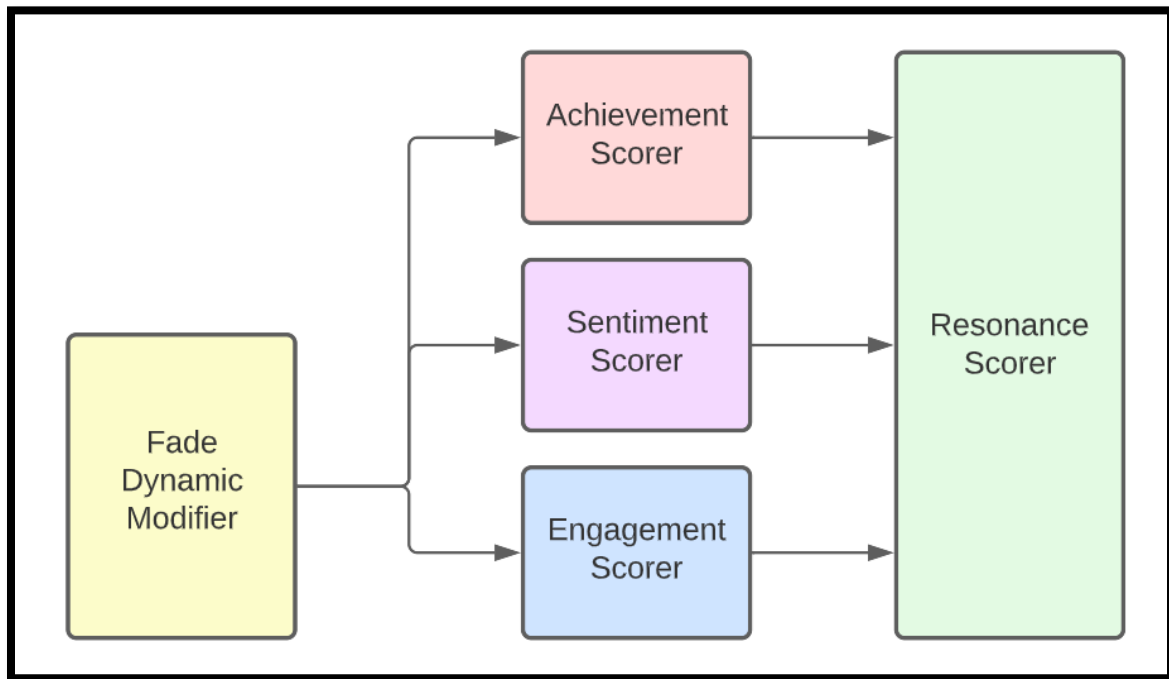


Figure 7: Fade Dynamic modifier being applied to scores used by resonance.

The new functionality is the trend enhancer. The trend enhancer feature identifies local trends in students' resonance scores and visualizes this information on the journey map, highlighting the information for the user. This feature identifies similarities in the changes in resonance values in the class's students and individual persona groups. The feature uses resonance scores very similarly to the clustering for persona groups, but here it identifies trends based on the bell curve of the total submissions over that period of time, positive, neutral, and negative. The feature is driven by command in the UI, relying on the user to set the time frame of search, this specifies to the program what achievement, sentiment, and engagement components to use in the trend search. The trend search will use changes in scores from time start to time end to highlight movement patterns of interest. An example would be to identify trends over the last two weeks of the semester, the trend enhancer would take the rates of change in scores of students over that time period to identify students who stayed the same, improved, and declined. The bell curve will normalize the rates of changes in resonance so both extremes and majority can be portrayed with the feature. The other example is when the same is applied to each persona group. Using the tool to track students positions in persona groups and how the persona groups react to new course components that impact resonance. This feature will also help us when testing the accuracy of persona groups, if the students in the group move together, this will provide great opportunities for the instructor to follow students in these groups and learn how they move between persona groups and why some students don't move to different persona groups. The feature has a second minor component that just highlights students who changed persona groups during the user selected time period, or whatever the defined time segments of the x-axis are in the journey map.

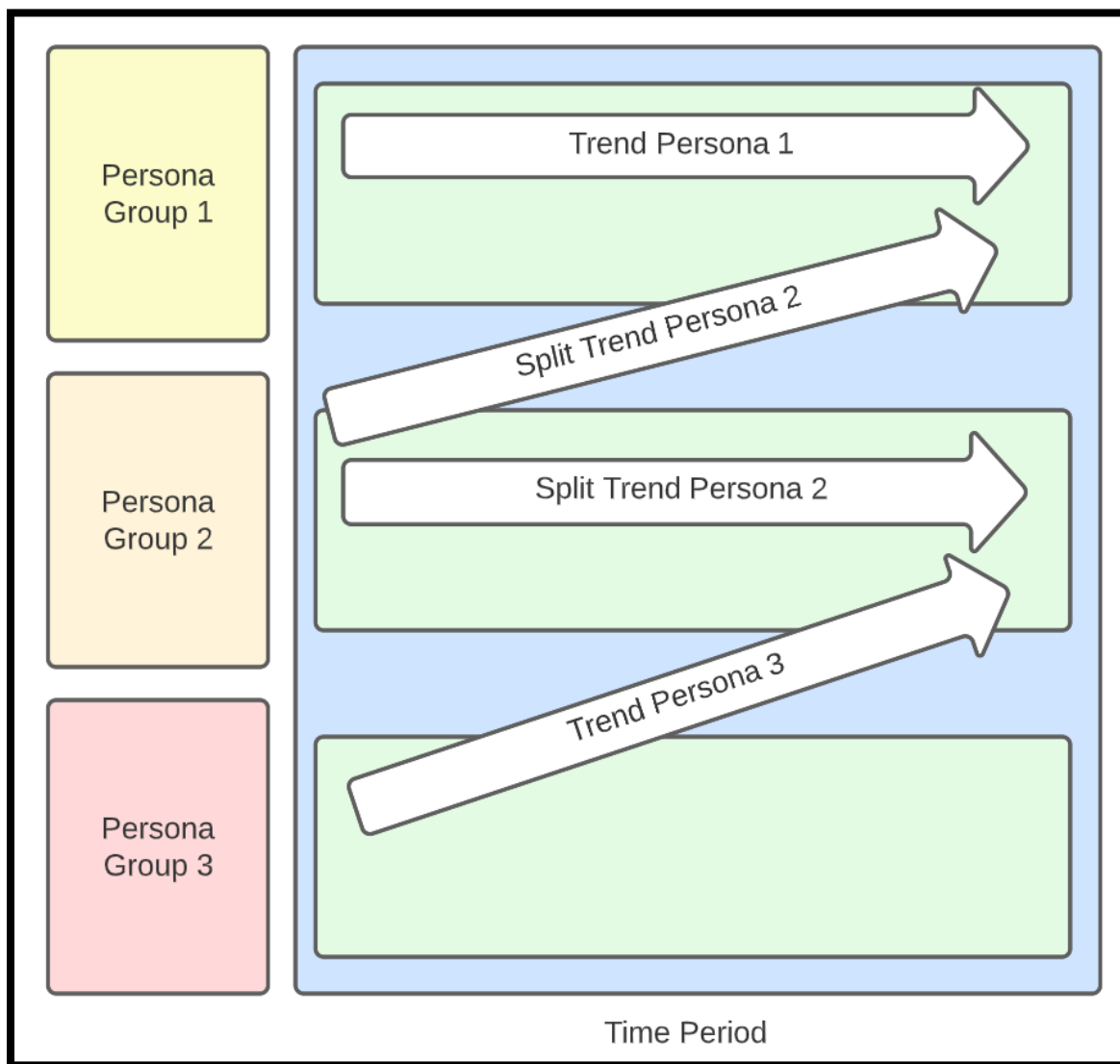


Figure 8: Visualization of identifying trends of students between persona groups.

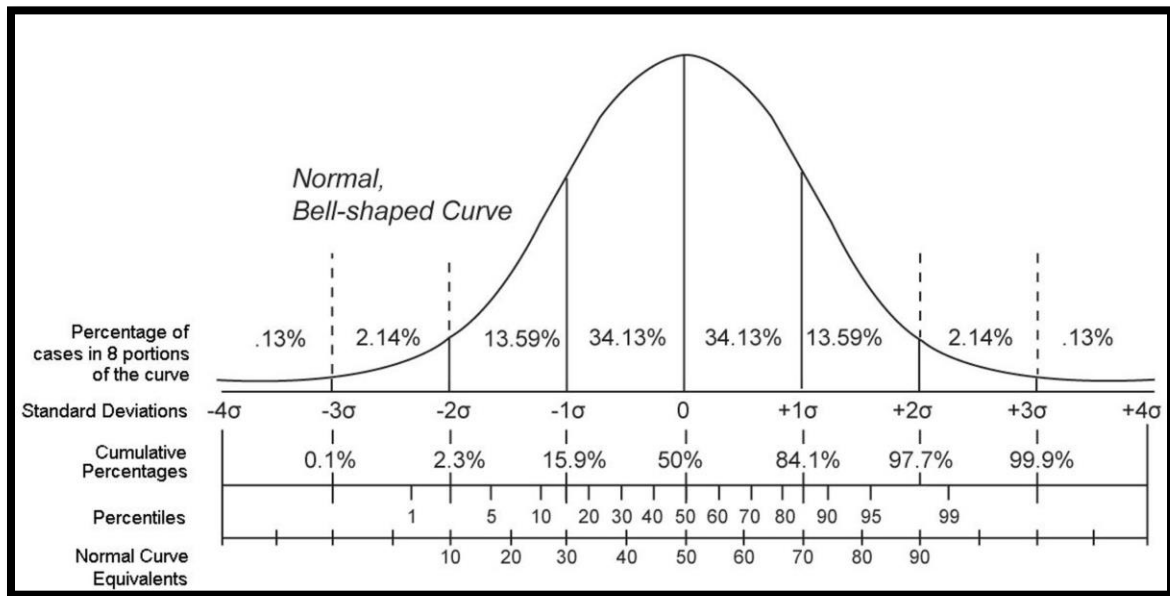


Figure 9: Bell curve example, used to select students of similar resonance movement. Taken from ImagineSchools.org.

Flexible persona groups were a feature implemented in the Mark I iteration, but was not refined to the point where it would work as intended and didn't meet the requirements of the client. The changes we are making aim to expand on the flexible capabilities and logic that expands the possibilities of the tool. Our implementation will provide real controls of the definition of personas to the user and incorporate the same pre-made filters and easy modification in the user interface just like resonance. We plan to adapt the new functionality to the previous team's persona implementation and use the same k-means clustering algorithm for grouping students into personas.

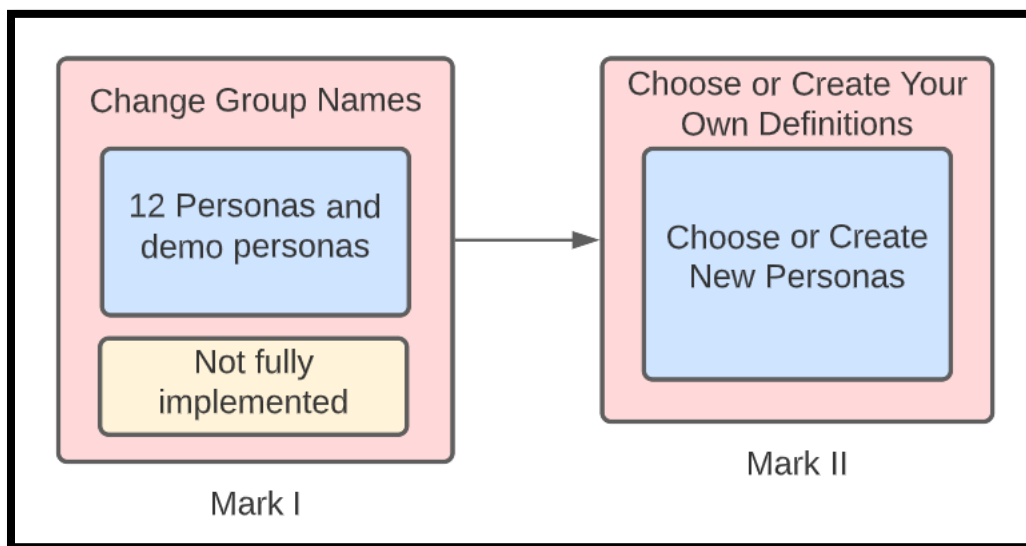


Figure 10: Mark I vs Mark 2 Persona implementation.

The UI changes will encompass all three goals of Mark II. The general usability and functionality of the application will be improved by putting the journey map front and center. The expanded journey map will allow for intuitive discovery of insights. The static top menu bar will provide access to common and other mandatory functions. Other menus will be fully customizable and can either be hovering over the journey map or framed around it. The new resonance and persona constructs and features will require efficient means of changing the weights of the selected elements that are used to measure resonance and visualized in the journey map.

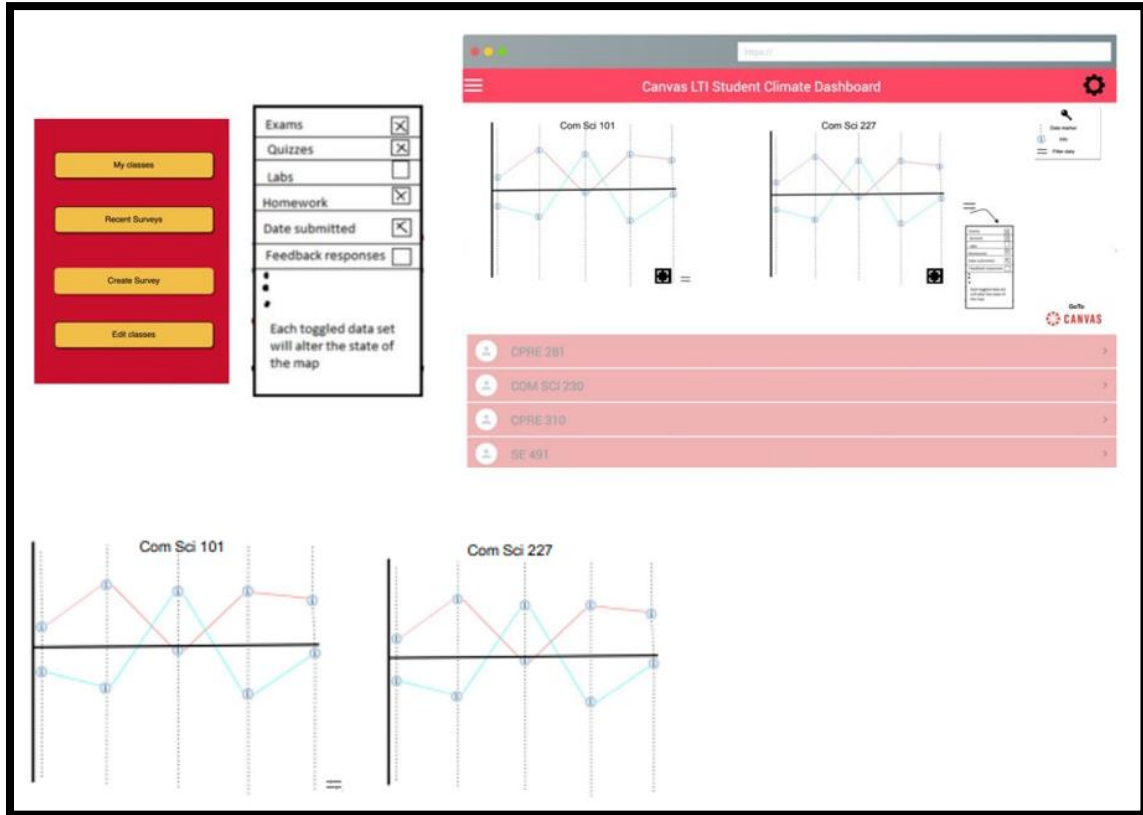


Figure 11: UI collage. Demonstrating Canvas integration, flexible position menus, and full screen capable journey maps. Taken from sddc21-19 Design Document Final Version.

The functionality of the application can be improved by meeting FERPA compliance. Being allowed to use the university's system will greatly decrease the work the instructor needs to do to set up the application for each course. The changes to the resonance and journey maps algorithms will need to be flexible in terms of allowing API control of template creation and through the user interface. The student facing functionality will be a feature disabled version of the full application, making it easy to adapt for the student view.

4.3.3 Functionality

Users will access the application through the course page on Canvas, same as Assignments, Grades, etc.

For the instructor:

- They can choose which students or groups to display on the journey map
 - Make calls to Canvas API to only get data for the options selected
 - Calculate the individual resonances from the data received from Canvas
 - Display the resonance on the journey map
- They can change the weights of assignments and resonance factors
 - Change the variable values when calculating the resonance value

All users (students and instructors):

- Can apply filters
 - There will be 4-6 default templates based on the most common users and their needs
 - Some will only be displayed to instructors
 - Users can also create their own templates and save the template to be used again
- Can export the journey map to image or pdf for download
- Can full screen the journey map
 - Needs to be dynamic for different technologies (mobile vs pc)

4.3.4 Areas of Concern and Development

Currently, our design satisfies most requirements and meets user needs. We have had a lot of communication with our client about their expectations and needs. We discussed these needs with our advisor to formulate a design that can be reasonably accomplished with the team's knowledge and experience. We understand the most important parts of the design that need to be focused on first and which parts can be put to the side in case we run into issues or run out of time. Our design expectations and timeline have been well communicated to the client and advisor and they both approve of the current plan. We are using an agile development process to account for any changes that need to be made.

We need to work with the university to figure out how to integrate with the university's Okta services. We need to integrate with their Okta service to get access to student academic data which is integral to all our functions. Since we are only accessing student data through the Canvas API, we must use the Okta authentication services to get a token to access the data. This is mostly important for the instructor's view because they are accessing data that is not their own. We could build a workaround for students when they log in, they give permission to share their data with the application so we can use it.

We are actively communicating with the university's registrar and IT personnel to get information on what is required of us as developers to gain access to academic data and what is needed for the technological integration to access the data. We will most likely need to complete FERPA training and be given specific roles within the university to access the data as developers. We are still waiting for feedback from the university for help with the Okta integration.

The initial questions we have for the university are: What steps are needed to integrate an external learning tool that accesses FERPA protected data? What standards or policies do we need to follow to manage and access the data?

4.4 Technology Considerations

To implement this project, we have looked at the previous team's design and see no major reasons that we will stray away from the technologies that they had decided on for their implementation. For this project and all the components that will go into it we will utilize a variety of languages and technologies to do so. Based on the previous team's design and our experience the main languages that will be utilized are Python, JavaScript, HTML and SQL. Additionally complying with FERPA regulations is also an important driver in designing this project. We can't store any student or instructor data, but rather must pull the data from Canvas we need to access the data. Ensuring that the application can run and get this data is very important to the implantation of this project.

In order to integrate all these different, it needs to be easily managed and to do this the use of containers will be critical to incorporate everything into the project and to comply with the security and regulatory requirements that are critical to the project. Docker is a very good application that can help provide the project with many tools for deployment and testing such as Kubernetes and CI/CD workflows for continuous delivery of the application. This along with the previous team's project will allow us to use preexisting resources to ease the development process ideally and ensure all the standards that must be maintained.

A possible weakness with this project is if it can be integrated within Iowa States Canvas. An alternative approach that we will have to take in this scenario would be to use a 3rd party Okta which would have to ask the students and professor for their Canvas tokens in order to use to application and have access to their data. This solution would work, but the drawbacks would be since there would be added security concerns here since then we would have to store different users' tokens in our database.

4.5 Design Analysis

We have been working with our Client and Advisor to gather features, functionality, and scope of application's changes and improvements. We anticipate that as we implement the proposed design changes, there will be revisions required to keep the application healthy and usable for its most pertinent use cases. We plan to iteratively test and reflect on the design changes and take the required steps to ensure we can meet the requirements of our client.

We have not yet built or implemented our design changes. We will be focusing on a prototype of the panned UI changes and student facing interface. The resonance and journey map models will have diagrams that describe how we plan to increase the flexibility when they are defined and metrics that can be used to gauge the accuracy of the algorithms. Keeping close tabs on these changes will help us adapt quickly to issues and find alternative solutions within the bounds of our goals.

5 Testing

Our team is determined to deliver to our client a fully functioning application that can be immediately implemented in a class setting where the client can use automatically generated journey maps to empathize with students and improve course design. We plan to make the most of industry standard tools and techniques to ensure our application meets the strict client and student expectations. We have planned the use of Unit, Interface, Integration, System, Regression, Acceptance, and Security testing for our application. If the application is to be successful and to be rolled out to additional faculty members or departments, the software must be able to handle the diverse operating conditions and user expectations to be a successful application. We hope that executing our testing plan will validate our designs and strengthen the foundation for great instructor and student insights.

The team does not have any software in development and has not had the opportunity to implement tests for the planned changes to the application. We look forward to using our guide to develop tests that validate the function of the application and verify the quality of the user experience.

5.1 Unit Testing

Unit testing will be a very important aspect of testing for this project. It will be important to do unit testing for the backend of the code which will be done using the PyUnit framework since we plan on using python for the majority of the backend. It is important to have these easy to run and update so that as new components get added the tests can be easily updated and ensure that the code is working. This is often the first level of testing just to ensure that certain components are operating correctly, and this will be especially important for other levels of testing as well. This will ensure that as additions are made to the code it can be ensured that nothing has broken and if so let us know what has happened to cause this test to be failed.

Specific use cases will primarily focus on different combinations of calls from the front end. For example, having a professor open a saved template, customizing the resonance model, or opening the window to inspect a particular data point. In each case, we have specified inputs. As the input is passed to each micro service, it branches out into multiple tests. For example, a change in the calculation for resonance will require that inputs and outputs for the three category scorers be confirmed. Clustering and final resonance score will also be evaluated on a unit level. Unit testing for the API gateway will primarily involve network calls and responses with appropriate parameters.

5.2 Interface Testing

One of the major focus points of the second iteration of the project is the improvement of the UI and the ability to provide flexibility when setting up resonance definitions and interacting with the journey map. The many planned changes of the UI will be more technical features and designs that will benefit from a well-executed interface testing plan. The first iteration sought to use Selenium for testing the UI and noted, CI/CD would be a challenge with strong UI testing. We could not find any evidence of the past team using a tool like Selenium for front end testing.

The team noted the testing of different user permissions and use case testing on popular web browsers. There may be challenges if the prior team were not successful in using tools beyond manual use case tests for the UI.

Our testing will focus on incorporating the Selenium Suit of tools for testing web browsers, OSs (Operating Systems), and scripts for automated use case testing. If Selenium is insufficient for our needs or we experience technical difficulties, resorting to manual testing, like the prior team, is a last resort. We plan to focus on the following use cases: operating systems, web browsers, Canvas apps, user profiles, classes, resonance builder, and journey map. These use cases are essential to the application's function and effectiveness as a research and student climate tool.

5.3 Integration Testing

Integration testing will confirm that multiple components are properly coordinating. For example, unit tests may ensure that each of the separate components accurately measure resonance for a specific category. The clustering and resonance components are highly dependent on the scoring services. Therefore, we should like to know that data is passing through dependencies as expected—and providing the desired result. However, since the API gateway will be directly coordinating the services, it will be sufficient to test only the integration between the gateway to and from each of the individual micro services, simplifying the dependencies in the backend to a relatively flat structure.

Integration testing which includes frontend and backend components shouldn't be necessary. So long as all responses to the UI are handled in testing, there's no benefit to conducting integration testing, as the dependency is simply a matter of each "end" operating properly.

Database and Canvas integration is unnecessary, as the handling of responses is straightforward and either fails or passes on a network level. However, the Canvas cache is a delicate component, and as it serves as a crucial intermediary between Canvas API and the application's API gateway, the tests for the cache should be robust, ensuring that data from the API is cached and returned to the API gateway as expected.

Because so much of the backend is hosted through Kubernetes, integration testing will require a service capable of building and tearing down the pods for testing specific services. Jenkins can run a testing pipeline for integration testing, using the Jenkins-Kubernetes-plugin. Ideally, Jenkins will deploy nodes for testing purposes, and we can confirm integration across the pods hosted within the nodes.

5.4 System Testing

Since we will be developing incrementally and ideally using test driven development, the planned unit tests, interface tests, and integration tests discussed above will provide the majority of our test coverage. System testing will be more focused on testing all areas work together overall. This will be better established once we really begin development and have the tools for front and back end better defined. As of right now, our current plans for all areas of testing will provide the coverage we need.

5.5 Regression Testing

We are ensuring that new additions do not break old functionality with CI/CD pipelines. It will be tested little by little and with every push after every session to make sure things still work correctly. Currently there aren't any critical features that we ensure not to break since we have not started building our project yet, but a big feature is the filtering feature. Our client has emphasized filtering information, so we must ensure it works nicely for him.

5.6 Acceptance Testing

We will demonstrate that both functional and nonfunctional requirements are being met by actively meeting requirements originally created and approved by our client. This will be measured through testing specifically made for these requirements. We involve our client in acceptance testing in our weekly meetings where we show new ideas and prototypes and adjust/change them with their input and wishes.

5.7 Security Testing

Most of our security testing will be focused on the Okta integration. We are speaking with the university's IT team to learn the necessary and best way to test the integration to make sure it is secure. Since our major security risks are associated with our third-party integrations, we will begin a better testing plan for this once the process for establishing those integrations is completed. Otherwise, we will test that our database is secure and is following proper safety and security protocols.

5.8 Results

Our tests are designed to meet the requirements of the users' operating environments. Tests will be available for all major components and cover all major use cases. Organization will be key in managing the tests and finding errors that prevent the application from fulfilling its goals. We will use the mentioned tools effectively to create safe and secure code.

6 Implementation

Users will access the application through the course page on Canvas, same as Assignments, Grades, etc.

For the instructor:

- They can choose which students or groups to display on the journey map
 - Make calls to Canvas API to only get data for the options selected
 - Calculate the individual resonances from the data received from Canvas
 - Display the resonance on the journey map according to filters and settings
 - Adapt to updated filters and settings
- They can change the weights of assignments and resonance factors
 - Change the variable values when calculating the resonance value
 - Reapply all filters and setting then display
- They can change the interface appearance of the journey map through text input of code

- Conversations still need to be had to decide how much the instructor will want and be able to change

All users (students and instructors):

- Can apply filters
 - There will be 4-6 default templates based on the most common users and their needs
 - Some will only be displayed to instructors
 - Users can also create their own templates and save the template to be used again
- Can export the journey map to image or pdf for download
- Can full screen the journey map
 - Needs to be dynamic for different technologies (mobile vs pc)

7 Professional Responsibility

7.1 Areas of Responsibility

Area of responsibility	Definition	IEEE Code of Ethics	Priority	Performance
Work Competence	Perform work of high quality, integrity, timeliness, and professional competence.	To maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations.	High	Medium
Financial Responsibility	Deliver products and services of realizable value and at reasonable costs.	To avoid real or perceived conflicts of interest whenever possible, and to disclose them to affected parties when they do exist.	NA	NA
Communication Honesty	Report work truthfully, without deception, and understandable to stakeholders.	To seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, to be honest and realistic in stating claims or estimates based on available data, and to credit properly the contributions of others.	Medium	High
Health, Safety, Well-Being	Minimize risks to safety, health, and	To hold paramount the safety, health, and welfare of the public.	Medium	High

	well-being of stakeholders.			
Property Ownership	Respect property, ideas, and information of clients and others.	NA	High	Medium
Sustainability	Protect environment and natural resources locally and globally.	To strive to comply with ethical design and sustainable development practices, to protect the privacy of others, and to disclose promptly factors that might endanger the public or the environment.	Low	Medium
Social Responsibility	Produce products and services that benefit society and communities	To improve the understanding by individuals and society of the capabilities and societal implications of conventional and emerging technologies, including intelligent systems.	High	High

7.2 Project Specific Professional Responsibility Areas

Work Competence:

High importance, because the complexity of the task and the desired outcome requires a fair amount of competence. We need to get up to speed with several API's, a complicated micro-service architecture, FERPA laws and university processes/administration, and OKTA integration. Because the project will be ongoing after our mark, we prioritize documentation and simplification, taking time to learn what's needed and lower the competence bar for the next team.

We've faced a lot of challenges with the infrastructure of the project and getting authorization. Being locked out of the existing OKTA, facing conflicts with plugin updates, and steep learning curves for Docker and Kubernetes have helped us consider ways to simplify and improve the application, but they're also slowed us down.

Financial Responsibility:

There are no finances involved, we're operating with materials and services provided by the university for a senior design team.

Communication Honesty:

Medium importance, because the project is important to the plans of the client, but the stakes for them are relatively low. We want the client to be aware of expected features and dates so they can plan to use the application in research and teaching. They also will want to consider areas to continue to develop the app in the future.

The clients and stakeholders are aware of the challenges and limitations of the team. Meeting weekly provides a chance to present challenges and come to terms with timelines, both on the client and developer side.

Health, Safety, Well-Being:

High importance, because we want to be conscientious about doing no harm. Some potential consequences to the mental health of users include:

- If the journey map provides incorrect calculations, it has negative effects on both students and professors. Students will receive guidance from the map which doesn't correlate with their performance. Professors will make decisions based on the maps—if the data calculations are erroneous, then their efforts could be counterproductive.
- An unintended consequence would be students using the journey map and experiencing negative reinforcement. We want to be sure to frame the journey maps as tools which can help students meet their potential.

We're conscientious of the effects the program could have, and we're planning ways to avoid potential unintended consequences. Thorough testing will help avoid errors, and proper framing of the user interface for students will help guide their experience and avoid unintended consequences. We also want to encourage our client to include beta release with the consent of students to measure and detect the consequences of the student-facing maps.

Property Ownership:

High importance, because we didn't develop the current state of the program, and we want to pass on a product that's in good order for the next owners. As university students working with professors, we have a shared stake in the application, and we're investing in a final product which improves course experience and performance.

We're careful to preserve the previous team's work while adapting it to our team's abilities and needs.

Sustainability:

Low importance, because the potential impact of the project is low. Data usage and performance are obviously concerns, but they're currently not high on the priority list.

We want the program to be efficient, however, it's not a top priority and we don't have specific plans to improve efficiency. It currently uses caching to limit API requests to Canvas, so the infrastructure is well set up.

Social Responsibility:

Medium importance, because we want to improve the culture and experience of students and professors at the university by offering a tool which enhances empathy and provides a customizable, chronological gauge of student engagement. It can also help to emphasize student performance with metrics besides grades.

User benefit and experience are integral to the planning process. In accordance with the client's needs and specifications, we've focused on designing a product which produces a high level of benefit and is easy to use.

7.3 Most Applicable Professional Responsibility Area

Social Responsibility: The primary goal of the product is to empower the users, which in turn will help demonstrate the app's potential to other users. Present and future students will benefit, as professors can gauge psychic limitations and design courses in sync with student needs. An intuitive interface, heavy customization features, quick loading time, and simple integration will provide a lasting tool with profound potential for university education.

8 Closing Material

8.1 Discussion

The main thing we have completed for our project is the Design Document that outlines our plans and approaches that will be implemented next semester. We are yet to get the previous team's application running correctly on our server. We were also unable to test any of our changes due to not being able to get the server to correctly run the past team's iteration of the project. We will need to have that sorted and catch up on what we missed out on this semester before next semester starts, otherwise we will not be able to meet our goals for our project. By the end of next semester, we plan to have a reworked user interface, flexible definition and accurate measurement of resonance, and a student facing interface.

8.2 Conclusion

This semester we have completed all assignments and components of the Design Document. We have gained an understanding of resonance, personas, and journey maps through the previous team's project, meeting with the client and advisor, and reading their published paper. We were unsuccessful in implementing the previous team's application on our server and testing our planned changes on it. We plan to meet the project's goals by acting on our plans defined in the Design Document. The goals we are aiming to achieve are for an improved user interface, flexible definition and increased accuracy of resonance and journey map modeling, and a student facing interface. Our plan incorporates agile components that are broken down into manageable modules with some flexibility, so when we are working on them, we can be adaptable while still moving forward. We are overall dissatisfied with our progress this semester and will need to come together to redefine our commitments to set ourselves up for success next semester.

8.3 References

- (2022, December 2). Retrieved from Imagine Schools: <https://www.imagineschools.org/wp-content/uploads/2016/10/Bell-Curve.jpg>
- IEEE-CS/ACM. (2022, November 4). *Code of Ethics*. Retrieved from Computer: <https://www.computer.org/education/code-of-ethics>
- Nickolas D. Fila, D. T. (2022). Considerations for the Use of Personas and Journey Maps in Engineering Course Design. *ASEE Excellence Through Diversity*. Minneapolis: ASEE.
- Rains, J. (2022, December 2). *Student Journey Mapping*. Retrieved from Rains Media: <https://www.jpains.com/student-journey-mapping/>
- sddec21-19. (2022, December 2). *Canvas LTI Student Climate Dashboard*. Retrieved from <https://sddec21-19.sd.ece.iastate.edu/docs.html>
- sddec21-19. (2022, December 2). *Design Document Final Version*. Retrieved from Senior Design Team sddec21-19: https://sddec21-19.sd.ece.iastate.edu/docs/Design_Document_Final_Version.pdf
- sddec21-19. (2022, December 2). *Project Plans*. Retrieved from Senior Design Team sddec21-19: https://sddec21-19.sd.ece.iastate.edu/docs/Senior_Design_Demo.mp4
- sddec21-19. (2022, December 2). *Project Poster*. Retrieved from Senior Design Team sddec21-19: https://sddec21-19.sd.ece.iastate.edu/docs/492-sddec21-19_poster.pdf
- “Contextualizing Professionalism in Capstone Projects Using the IDEALS Professional Responsibility Assessment”, *International Journal of Engineering Education* Vol. 28, No. 2, pp. 416–424, 2012

8.4 Appendices

8.4.1 Team Contract

Team Members:

- | | |
|-------------------|--------------------|
| 1) Howard Chi | 2) Joshua Harvey |
| 3) Colin Hasbrook | 4) Jonathan Giblin |
| 5) Hailee Leonard | 6) Elias Simpson |

Team Procedures

1. Day, time, and location (face-to-face or virtual) for regular team meetings:
 - a. Mondays 3:30 – 4:30 over Microsoft teams with advisor and client.
 - b. Wednesdays 3:00 – 4:30 over Discord with team members.
2. Preferred method of communication updates, reminders, issues, and scheduling (e.g., e-mail, phone, app, face-to-face):
 - a. Meetings will be conducted virtually over Microsoft Teams and Discord
3. Decision-making policy (e.g., consensus, majority vote):
 - a. Decisions are made by majority vote
4. Procedures for record keeping (i.e., who will keep meeting minutes, how will minutes be shared/archived):

Participation Expectations

1. Expected individual attendance, punctuality, and participation at all team meetings:
 - a. Come prepared to meet all personal and team requirements and duties at each team meeting.
 - b. Be on time, focused, and respect advisor and client.
2. Expected level of responsibility for fulfilling team assignments, timelines, and deadlines:
 - a. If a task or responsibility is accepted by a team member, fair effort should be put forth to complete the task according to concurrent tasks and deadlines. If the tasks can't be finished when expected, it should be communicated with the team and steps should be taken to step in and/or adjust plans accordingly.
3. Expected level of communication with other team members:
 - a. Respond to questions in a timely manner. If you are running into an issue let the group know so that we can work through it as a team.
 - b. Communication should be sent to all members when necessary. Member to member communication is fine when content doesn't concern other members.
4. Expected level of commitment to team decisions and tasks:
 - a. It is expected that each member takes responsibility for their assigned tasks and commitment is sufficient to match the team's progress.

Leadership

1. Leadership roles for each team member (e.g., team organization, client interaction, individual component design, testing, etc.):

- a. Product Owner: Henry Duwe and Nicholas Fila
- b. Project Manager: Josh Harvey
- c. Client Interaction: Josh Harvey and Team
- d. Individual Component Design:
 - a. UI – Howard Chi
 - b. UX – Hailee Leonard
 - c. Resonance – Josh Harvey
 - d. Journey Map – Josh Harvey
 - e. Database – Colin Hasbrook
 - f. Security – Jon Giblin
 - g. Login Authentication – Jon Giblin
 - h. CI/CD – Hailee Leonard
 - i. Server – Elias Simpson
- e. Testing: Everyone

2. Strategies for supporting and guiding the work of all team members:

- a. Weekly Meetings
- b. Jira Board for tracking issues and tasks
- c. Cooperation and team spirit to resolve challenges

3. Strategies for recognizing the contributions of all team members:

- a. Jira will be used for issue and project tracking

Collaboration and Inclusion

1. Describe the skills, expertise, and unique perspectives each team member brings to the team.

Josh Harvey – Team management, organization, software engineer, requirements, design.

Colin Hasbrook - Software Engineering, MBA; data analysis (python), front end development

Jonathan Giblin - Software engineer, back end.

Elias Simpson – SE, full stack, cloud development, management, vision.

Hailee Leonard – Software engineer: front end, API, design, requirements.

Howard Chi – data engineer

2. Strategies for encouraging and support contributions and ideas from all team members:

- a. Safe environment for encouraging and supporting contributions and ideas.
- b. Focus on positives

3. Procedures for identifying and resolving collaboration or inclusion issues (e.g., how will a team member inform the team that the team environment is obstructing their

opportunity or ability to contribute?)

- a. If you have an issue let the team know through discord and the team will work through the issue either as an entire group or in small groups where the issue pertains.

Goal-Setting, Planning, and Execution

1. Team goals for this semester:

- a. Have fun and enjoy the experience.
- b. Develop technical and soft skills.
- c. Good team organization and communication
- d. Able to define improvements that we plan to achieve in the next two semesters.
- e. Achieve as much of if not more than what we have planned.
- f. Stay motivated during the entire project.
- g. Create a final product that can be used by students to measure their progress and improve their satisfaction from ISU courses.
- h. Create a final product that can be used by ISU faculty and staff to best design, run, and improve courses to best meet the needs of students and university goals.

2. Strategies for planning and assigning individual and teamwork:

- a. Use tickets to plan and assign work using Jira.
- b. Each team member will be allowed to choose their tasks that best fit their skills and ambitions.
- c. A separate document will be used to document all tasks, goals, and progress for the semester.
- d. After the next meeting, we will have a better understanding of what tasks we will be doing and how to best plan and assign. We will follow guidelines as outlined in class.

3. Strategies for keeping on task:

- a. Jira board, agile environment
- b. Class deadlines
- c. Friendly reminders

Consequences for Not Adhering to Team Contract

1. How will you handle infractions of any of the obligations of this team contract?

- a. Group intervention, try to resolve issues within team.
- b. Go to Henry or Nick if it can't be resolved from team intervention.

2. What will your team do if the infractions continue?

- a. Remove person(s) from team if can't be resolved within team or advisor.

a) *I participated in formulating the standards, roles, and procedures as stated in this contract.*

b) *I understand that I am obligated to abide by these terms and conditions.*

c) *I understand that if I do not abide by these terms and conditions, I will suffer the consequences as stated in this contract.*

1) Colin Hasbrook DATE: 9/12/2022

2) Howard Chi DATE: 9/12/2022

3) Josh Harvey DATE: 9/12/2022

4) Hailee Leonard DATE: 9/12/2022

5) Elias Simpson DATE: 9/13/2022

6) Jonathan Giblin DATE: 9/16/2022