



## ***COMP3850 Project Deliverable Certificate***

Name of Deliverable	<i>Deliverable 4</i>
Date Submitted	<i>18 / 05 / 2022</i>
Project Group Number	<i>18</i>
Rubric stream being followed for this deliverable (highlight one) <i>Note: the feasibility study has the same rubric for all streams.</i>	<b><i>SOFTWARE Rubric</i></b> <b><i>GAMES Rubric</i></b> <b><i>CYBERSECURITY Rubric</i></b> <b><i>DATA SCIENCE Rubric</i></b>

We, the undersigned members of the above Project Group, collectively and individually certify that the above Project Deliverable, as submitted, **is entirely our own work**, other than where explicitly indicated in the deliverable documentation.

INITIALS	SURNAME	GIVEN NAME	STUDENT NUMBER	SIGNATURE (IN-PERSON OR DIGITAL)
EH	Horvath	Erik	46042628	
ST	Torfeh Nejad	Sepehr	46042547	
RS	Shrestha	Rojwal	46137572	
MI	Ikeda	Marcus	45415064	
LT	Te	Lance	45581819	

*NB: please write all details clearly (if handwritten).*

© Macquarie University, 2021

**List of tasks completed for the deliverable and activities since last deliverable certificate with totals for each individual team member and whole team**

Performed by <i>(Student Names)</i>	Duration <i>(hrs)</i>	Complexity <i>(L, M, H)</i>	Name of task	Checked by <i>(Initials)</i>
<b>Lance Te</b>	10	M	Writing my sections in D4 and editing others' sections	ST, EH
	3	M	Formatting D4	
	15	H	Implemented feedback for MVP (refactoring JS, refactoring CSS, loading page, login page, mainly front-end)	ST
	1	L	MongoDB changes	ST
	5	L	Weekly team and client meeting	
	3	M	Client meetings + presentations	
	3	L	Revised Testing Doc	ST
	5	H	Revised all D4 section by section, comment by comment	ST
<b>Total</b>	45			
<b>Sepehr Torfeh Nejad</b>	5	L	Weekly team and client meeting	
	2	M	Client meetings + presentations	
	5	H	Implementing new routes for the backend survey	LT
	5	H	Implementing feedbacks that we got from the client (backend side and redux)	LT
	4	H	Changing the MongoDB schema based on the feedback	LT
	5	H	Writing my section of the D4	EH
	5	M	Code review on Github for pull request	LT
	2	H	Revise user Manual	LT
<b>Total</b>	33			
<b>Erik Horvath</b>	6	M	Revising D3 sections from feedback	LT
	3	L	Attending weekly team meeting	LT
	1	L	Weekly client meetings	LT
	5	H	Adding content to D4	ST
	3	M	D4 research	ST
	2	L	Team communication and general organisation	ST
	4	H	Creating new test cases for D4	RS
<b>Total</b>	24			

Performed by <i>(Student Names)</i>	Duration <i>(hrs)</i>	Complexity <i>(L, M, H)</i>	Name of task	Checked by <i>(Initials)</i>
<b>Marcus Ikeda</b>	5	M	User Manual	LT
<b>Total</b>	5			
<b>Rojwal Shrestha</b>	5	M	Writing And Updating Scoping Doc	LT, ST
	5	M	Updating Project Plan	LT, ST
	5	M	Editing and updating D3 Timeline and Schedule	LT, ST
	5	M	Updating And Revising Analysis Doc	LT, ST
	5	M	Updating and Revising Design Doc	LT, ST
	3	L	Team And Sponsor Meetings	LT, ST
<b>Total</b>	28			
<b>Team Total</b>	135			

## Revision Table

Project Plan		
Document Version	Description	Date
Project Plan v1.0	Original submission for Deliverable 2	31/03/22
Project Plan v2.0	Some work was carried out during the first week of holidays, updated timelines & turnaround expectations	12/04/22
Project Plan v3.0	Added project scope	11/05/22
Project Plan v3.1	Revised project schedule	12/05/22
Project Plan v3.2	Revised project timeline	13/05/22
Project Plan v3.3	Updated assumptions	14/05/22
Project Plan v3.4	Added Handover Requirements	15/05/22
Scoping Doc (SRS)		
Document Version	Description	Date
Scoping Doc v1.0	Original submission for Deliverable 2	31/03/22
Scoping Doc v2.0	Added a 'Contextual Diagram' to show the Onboarding UI in context of the entire Faethm application	15/04/22
Scoping Doc v2.0	Listed the definitions in alphabetical order	15/04/22
Scoping Doc v2.1	Considered user classes with accessibility requirements	16/04/22
Scoping Doc v2.1	Clarified front-end user browser support	16/04/22
Scoping Doc v2.2	Clarified what type of documents will be provided	17/04/22
Scoping Doc v2.2	Numbered each sub dot-point	17/04/22
Scoping Doc v2.2	Added 'Fit Criteria' for each functional and non-functional requirement	17/04/22
Scoping Doc v3.0	Added Functional Requirement 5 (FR5) – Landing Page	14/05/22
Scoping Doc v3.1	Removed section "3.2.2 – Cost" since no cost is involved	15/05/22

Analysis Doc		
Document Version	Description	Date
Analysis Doc v1.0	Original submission for Deliverable 3	28/04/22
Analysis Doc v2.0	Moved 'Client Feedback' from the Testing Doc to immediately after the MVP	10/05/22
Analysis Doc v2.0	Added 'Client Feedback' into each section and relevant action	10/05/22
Analysis Doc v2.1	Added a link to our GitHub repository	12/05/22
Analysis Doc v2.1	Added a 'Landing Page' feature	12/05/22
Analysis Doc v2.1	Added a 'Navigation Bar' feature	12/05/22
Analysis Doc v2.2	Added an explanation of the survey mechanic	13/05/22
Analysis Doc v2.3	Added an explanation of our data hosting solution (Mongo Atlas)	14/05/22
Analysis Doc v2.3	Added screenshots of our software hosting solution (GitHub)	14/05/22
Analysis Doc v2.4	Included latest feedback from Shu Tu (Faethm front-end dev)	15/05/22
Analysis Doc v2.4	Added 'Team Response' to feedback	15/05/22
Analysis Doc v2.5	Updated Use Case Diagram	15/05/22
Analysis Doc v2.6	Added Landing Page use-case	15/05/22
Analysis Doc v2.7	Added Week 11 meeting client feedback	18/05/22
Design Doc		
Document Version	Description	Date
Design Doc v1.0	Original submission for Deliverable 3	28/04/22
Design Doc v2.0	Added Landing Page to User Interface Layouts	08/05/22
Design Doc v2.1	Updated the Package Diagram to reflect the Landing Page	09/05/22
Design Doc v2.2	Revised ER Diagram	10/05/22
Design Doc v2.3	Added screenshots of our MongoDB	11/05/22
Design Doc v2.4	Added explanations of the data being collected and how it will be used	13/05/22
Design Doc v2.5	Updated the Window Navigation Diagram with the Landing Page	17/05/22

Testing Doc		
Document Version	Description	Date
Testing Doc v1.0	Original submission for Deliverable 3	28/04/22
Testing Doc v2.0	Moved 'Client Feedback' to the MVP in the Analysis Doc	14/05/22
Testing Doc v2.1	Added numerous 'User Interface' test specifications that tests the functionality of the UI	15/05/22
Testing Doc v2.2	Added 'User Experience' test specification that tests the user-friendliness of the UX	16/05/22
User Manual		
Document Version	Description	Date
User Manual v1.0	Original submission for Deliverable 4	18/05/22



# Project Plan

Lance Te, Sepehr Torfeh Nejad, Marcus Ikeda, Erik Horvath, Rojwal Shrestha

FÆTHM

## Contents

<b>1.</b>	<b>Hand Over Requirements.....</b>	<b>2</b>
<b>2.</b>	<b>Introduction .....</b>	<b>2</b>
<b>3.</b>	<b>Statement of Purpose .....</b>	<b>2</b>
<b>4.</b>	<b>Risk Management.....</b>	<b>3</b>
<b>5.</b>	<b>Resource Management.....</b>	<b>4</b>
5.1.	People .....	4
5.2.	Software .....	4
5.3.	Hardware .....	4
5.4.	Clients .....	4
<b>6.</b>	<b>Team Organisation .....</b>	<b>5</b>
<b>7.</b>	<b>Project Schedule.....</b>	<b>6</b>
7.1.	Task / Activities / Phases .....	8
7.2.	Timeline .....	11
7.3.	Resources Allocated .....	13
<b>8.</b>	<b>Process Model.....</b>	<b>13</b>
<b>9.</b>	<b>Documentation .....</b>	<b>14</b>
<b>10.</b>	<b>Assumptions .....</b>	<b>14</b>

## **1. Hand Over Requirements**

The client expects us to develop a landing page and other features as discussed in the project scope. All the code, scripts and documentation leading to conception of the final application is requested by the client at the end of project. When the code for the final product is completed and tested, we will hand over all the source code and test data which is present in our GitHub along with our MongoDB database. The code will have no problem integrating to the main Faethm base as it is developed in their preferred stack, MERN. A Google Drive folder which includes the project plan, SRS, User Manual and many other documentations maintained throughout the development will also be handed over. The test strategy for the application along with the test schedule and tools will be presented in the Testing document which is also uploaded to the google drive. Any further developments are to be uploaded onto the respective cloud services mentioned above, that is, Google Drive and GitHub.

## **2. Introduction**

Faethm AI is a SaaS based AI powered platform that uses extensive datasets to provide insightful information to professionals and their industry. The team at Faethm has partnered up with Macquarie University to provide the opportunity for students to gain valuable industry experience. The project tasked to us is to create a personalised landing page for a new user to improve their onboarding experience.

## **3. Statement of Purpose**

This project plan is the formal documentation providing an elaborate overview of the management and execution of the project as a whole. It is an important reference tool for all parties involved including the developers (Macquarie University students) and stakeholders (Faethm team). The project plan provides information about the scope of the project as well as the technical and management strategies of the project. This plan should be the primary resource for confirmation on the strategy for managing the project.

The scope of this project is to revamp the current Faethm landing page/dashboard and provide functionalities as follows:

- A survey is to be prompted if the user is signing in for the first time
- Categorize the user into pre-defined personas based on the survey
- Based on the persona customize the user's dashboard

Project areas defined in this project plan include:

- Risk management strategies including a risk matrix outlining the risk probabilities, consequences, and the potential impact of the associated risk
- Mitigation strategies to help avoid the risks or reduce the overall impact of the potential risk
- Management of the project resources including people, hardware software and other resources used in the project
- Organisation of the group members into roles and responsibilities
- Scheduling of all tasks and activities
- Identifying the products and documentation required for the project and justification for its purpose
- Software Development Life Cycle
- Gantt Chart with planned tasks, timelines, deliverables, and resource allocations for the project

## 4. Risk Management

It is important to mitigate the risk of undertaking a project by planning for potential risks that may occur. By following a risk matrix, we can assess the severity of a potential risk as well as how important it is to mitigate that risk.

	Negligible	Minor	Moderate	Significant	Severe
Very Likely	Medium	Medium	Medium/High	High	High
Likely	Low	Medium	Medium	Medium/High	High
Possible	Low	Medium	Medium	Medium/High	Medium/High
Unlikely	Low	Medium	Medium	Medium	Medium/High
Very Unlikely	Low	Low	Medium	Medium	Medium

Risk	Description	Likelihood	Impact
Impact to Current System	The project causes problems to the current system and as a result it no longer functions.	Unlikely	Significant
Loss of Data	The project does not run correctly and causes Faethm's data to be lost.	Very Unlikely	Severe
Changes to Requirements	Faethm changes project requirements creating more work that needs to be done	Possible	Minor
Team Unable to Complete Work	A member of the team is not able to complete their allocated work	Possible	Moderate
Project Incomplete	The project is not finished within the development period	Possible	Minor
Bugs in Software	Software has bugs during development	Very Likely	Negligible

Risk	Mitigation Strategy
Impact to Current System	The system should be reverted back to a state where it was previously functioning or a state from before the project was started.
Loss of Data	Data should be backed up before running the software.
Changes to Requirements	Maintain good organisation and communication with Faethm.
Team Unable to Complete Work	Maintain good communication between team members and have someone else take over their work.
Project Incomplete	N/A.
Bugs in Software	Debugging.

## 5. Resource Management

The resources required for the successful execution of the project are outlined here. The main resources are broken down into three categories: people, hardware and software. Since this is a software development project with the Faethm AI application being a SaaS hosted platform, the need for physical resources such as material goods and capital expenses is eliminated. Further elaboration on the three project resources is provided below.

### 5.1. People

This resource consists of two categories including the developers and the Faethm team. The developers are the 5 members from COMP3850 Group 18:

- Lance Te
- Erik Horvath
- Sepher Torfeh Nejad
- Marcus Ikeda
- Rojwal Shrestha

### 5.2. Software

The Onboarding UI project will be built using the MERN (MongoDB, Express, React, Node) stack for ease of integration with the rest of the Faethm application. The different layers of the UI will be developed using technologies across the MERN stack:

- Frontend: React.js
- Server: Express.js, Node.js
- Database: MongoDB

### 5.3. Hardware

All members require their own computer set up with access to the internet to be able to contribute to the project.

### 5.4. Clients

The Faethm team is primarily involved to provide guidance and support for the development team. Since they are the primary stakeholders as well as providing us access to speak with members from their development team, they are both a technical and strategic, high level, resource. The lead from the Faethm team is Mikhail Thornail who oversees the project and is our main source of guidance and feedback. The rest of the Faethm team includes Shu Tu and Karin Uli who particularly assists us on our frontend and backend developments, and design respectively.

## 6. Team Organisation

The members of COMP3850 Group 18 are broken up into specific roles to take advantage of team members strengths and create a clear delegation of duties. The following is a list of each member and their project role:

- **Lance Te:** Team Lead, Lead Formatter, Developer, Documentation Support
- **Sepher Torfeh Nejad:** Lead Developer, Documentation Support
- **Erik Horvath:** Editor, Documentation Officer, Developer
- **Marcus Ikeda:** Documentation Officer, Developer
- **Rojwal Shrestha:** Documentation Officer, Developer

These roles are distributed based on the agreed skillsets of the group members. The roles have clearly defined duties and the role descriptions outlined below:

- **Team Lead:** Main point of contact, in charge of all deliverables and project component submissions, organising group meetings and general team leader duties.
- **Lead Formatter:** Responsible for compiling all group contributions of deliverables and formatting documentation for presentation purposes.
- **Lead Developer:** Most experienced developer with ability to delegate tasks and advise other developers.
- **Lead Editor:** Looks for misspellings, incorrect grammar, missed punctuations, inconsistencies and general flow of the text.
- **Documentation Officer:** Responsible for creating the documentation and fill in all required information needed to satisfy deliverables.
- **Developer:** Responsible for the coding and development of the project.
- **Documentation Support:** Provide support to Documentation Officers with technical documentation requirements.

The Faethm Team currently consists of 3 members who have established skill sets in various fields and also assists us on different parts of the project:

- **Mikhail Thornail:** Team lead and main contact of the Faethm Team.
- **Shu Tu:** Reviews our code and provides feedback, guidance and necessary resources to drive us toward the envisioned application.
- **Karin Uli:** Reviews the application from a design perspective.

## 7. Project Schedule

ID	Task Name	Product	Days	Start	Finish
1	Deliverable 1	Feasibility Study	7	Wed 3/2/22	Thu 3/10/22
2	Deliverable 2	Project Plan and Requirements/Scoping Document	12	Wed 3/16/22	Thu 3/31/22
2.1	D2A1	Project Plan for Deliverable 2	4	Wed 3/23/22	Sun 3/27/22
2.3	D2A2	Scoping Doc for Deliverable 2	7	Wed 3/16/22	Thu 3/24/22
2.4	D2A3	Quality Manual for Deliverable 2	5	Sun 3/27/22	Thu 3/31/22
3	Deliverable 3	Update D2, MVP, Design, Test Cases	17	Wed 4/6/22	Thu 4/28/22
3.1	D3A1	Revised Project Plan	7	Sat 4/16/22	Sun 4/24/22
3.2	D3A2	Revised Scoping Doc	9	Wed 4/13/22	Sun 4/24/22
3.3	D3A3	Prototype	14	Wed 4/6/22	Sun 4/24/22
3.4	D3A4	Analysis Doc	9	Wed 4/13/22	Sun 4/24/22
3.5	D3A5	Design Doc	9	Wed 4/13/22	Sun 4/24/22
3.6	D3A6	Testing Doc	9	Wed 4/13/22	Sun 4/24/22
4	Deliverable 4	Update D2, D3 and User Manual	12	Wed 5/4/22	Thu 5/19/22
4.1	D4A1	Revised Project Plan	9	Wed 5/4/22	Sun 5/15/22
4.2	D4A2	Revised Scoping Doc	9	Wed 5/4/22	Sun 5/15/22
4.3	D4A3	Revised Assumptions	9	Wed 5/4/22	Sun 5/15/22
4.4	D4A4	Revised Analysis Doc	9	Wed 5/4/22	Sun 5/15/22
4.5	D4A5	Revised Design Doc	9	Wed 5/4/22	Sun 5/15/22
4.6	D4A6	Revised Test Doc	9	Wed 5/4/22	Sun 5/15/22
4.7	D4A7	User Manual	9	Wed 5/4/22	Sun 5/15/22
4.8	D4A8	Prototype	9	Wed 5/4/22	Sun 5/15/22
5	Deliverable 5	Final Group Reflective Report	12	Wed 5/18/22	Thu 6/2/22
6	Deliverable 6	Project Presentation	12	Wed 5/18/22	Thu 6/2/22
7	Deliverable 7	Final Web Application Delivery	11	Wed 5/18/22	Wed 6/1/22

<b>8</b>	<b>Deliverable 8</b>	<b>Final Exam</b>	<b>1</b>	<b>Tue 6/7/22</b>	<b>Tue 6/7/22</b>
<b>9</b>	<b>Checkpoint 01</b>	<b>Individual Contribution Form Submission (First Half)</b>	<b>4</b>	<b>Wed 3/30/22</b>	<b>Mon 4/4/22</b>
<b>10</b>	<b>Checkpoint 02</b>	<b>Individual Contribution Form Submission (Second Half)</b>	<b>4</b>	<b>Wed 6/1/22</b>	<b>Sat 6/4/22</b>
<b>11</b>	<b>Functional Requirement 1</b>	<b>Login (FR1)</b>	<b>11</b>	<b>Wed 4/6/22</b>	<b>Wed 4/20/22</b>
11.1	FR1A1	FR1 Login Functionality	4	Wed 4/6/22	Sat 4/9/22
11.2	FR1A2	FR1 Error/Unsuccessful Login Handling	4	Sat 4/9/22	Wed 4/13/22
11.3	FR1A3	FR1 Questionnaire Display	4	Wed 4/13/22	Sat 4/16/22
11.4	FR1A4	Testing for FR1	3	Sat 4/16/22	Tue 4/19/22
<b>12</b>	<b>Functional Requirement 2</b>	<b>User Persona (FR2)</b>	<b>11</b>	<b>Wed 4/20/22</b>	<b>Wed 5/4/22</b>
12.1	FR2A1	FR2 Persona Classification	4	Wed 4/20/22	Sat 4/23/22
12.2	FR2A2	FR2 Pre-defined Persona Display	3	Sat 4/23/22	Tue 4/26/22
12.3	FR2A3	FR2 User Input Capture Functionality	3	Tue 4/26/22	Thu 4/28/22
12.4	FR2A4	FR2 Store User Input Functionality	3	Thu 4/28/22	Mon 5/2/22
12.5	FR2A5	Testing for FR2	2	Mon 5/2/22	Tue 5/3/22
<b>13</b>	<b>Functional Requirement 3</b>	<b>Use Case (FR3)</b>	<b>5</b>	<b>Wed 5/4/22</b>	<b>Tue 5/10/22</b>
13.1	FR3A1	FR3 User's Use Case Determination from Answers/From FR2	4 days	Wed 5/4/22	Sat 5/7/22
13.2	FR3A2	Testing for FR3	3	Sat 5/7/22	Tue 5/10/22
<b>14</b>	<b>Functional Requirement 04</b>	<b>Personalisation (FR4)</b>	<b>9</b>	<b>Sun 5/8/22</b>	<b>Wed 5/18/22</b>
14.1	FR4A01	FR4 Welcome Functionality	4	Sun 5/8/22	Wed 5/11/22
14.2	FR4A02	FR4 Relevant Features Display	5	Wed 5/11/22	Tue 5/17/22
14.3	FR4A03	Testing for FR4	3	Sat 5/14/22	Tue 5/17/22
<b>15</b>	<b>Product Testing</b>	<b>Testing the Onboarding UI</b>	<b>10</b>	<b>Wed 5/18/22</b>	<b>Tue 5/31/22</b>

## **7.1. Task / Activities / Phases**

In order to deliver the project on time and to balance the workload amongst the team members, the project can be divided into tasks and activities.

### **1. Deliverable 1**

Summary: The task is to write a Feasibility Study to determine the practicality of the project based the team's capability and experience as well as the project functional requirements.

Participants: All members

### **2. Deliverable 02**

#### **2.1. D2A1: Project Plan**

Summary: Write a document specifying risk and resource management also including the project schedule

Participants: Erik, Marcus, Roj

#### **2.2. D2A2: Quality Manual**

Summary: Write a document showing how quality is assessed and maintained throughout the project

Participants: All Members

#### **2.3. D2A3: Scoping Document**

Summary: Write a document providing an overview of the context and functionality of the project

Participants: Sepehr, Lance

### **3. Deliverable 3**

#### **3.1. D3A1: Revised Project Plan**

Summary: Revise and update the previously written document based on the developed MVP/Prototype.

Participants: Roj

#### **3.2. D3A2: Revised Scoping Doc**

Summary: Revise and update the previously written document based on the developed MVP/Prototype.

Participants: Roj

#### **3.3. D3A3: Prototype / MVP**

Summary: Develop a minimum viable product and document its features and functionalities.

Participants: Lance, Sep

#### **3.4. D3A4: Analysis Doc**

Summary: Write a document to outline the use cases of the project.

Participants: Roj

#### **3.5. D3A5: Design Doc**

Summary: Write a document illustrating the system design.

Participants: Marcus, Lance, Erik

#### **3.6. D3A6: Testing Doc**

Summary: Write a document that specifies the testing strategies to be used on the MVP.

Participants: Lance, Sep

### **4. Deliverable 4**

#### **4.1. D4A1: Revised Project Plan**

Summary: Revise and update the previously written document based on the developed MVP.

Participants: Roj

**4.2. D4A2: Revised Scoping Doc**

Summary: Revise and update the previously written document based on the developed MVP.

Participants: Roj

**4.3. D4A3: Revised assumptions**

Summary: Revise the assumptions made for the project.

Participants: Roj

**4.4. D4A4: Revised Analysis Doc**

Summary: Update the previous version of the doc with the corresponding comments provided.

Participants: Roj

**4.5. D4A5: Revised Design Doc**

Summary: Update the previous version of the doc with the corresponding comments provided.

Participants: Roj

**4.6. D4A6: Revised Testing Doc**

Summary: Update the previous version of the doc with the corresponding comments provided.

Participants: Roj

**4.7. D4A7: User Manual**

Summary: Write a doc that showcases all the functionalities and features of the application to aid a new user.

Participants: Marcus

**4.8. D4A8: Prototype**

Summary: Improve on the already developed prototype.

Participants: Lance, Sep

**5. Deliverable 05**

Product: Final Group Reflective Report

Summary: a document reviewing all aspects of the project

Participants: All Members

**6. Deliverable 06**

Product: Project Presentation/Demonstration

Summary: the group will present their system to the sponsors, academics and students.

Participants: All members

**7. Deliverable 07**

Product: Final Group Reflective Report

Summary: a document reviewing all aspects of the project.

Participants: All members

**8. Deliverable 08**

Product: Final Exam

**9. Checkpoint 01**

Product: Individual Contribution Form (First Half)

Summary: Each member fills out a form assessing other member's contributions in the project.

Participants: All members

**10. Checkpoint 02**

Product: Individual Contribution Form (Second Half)

Summary: Each member fills out a form assessing other member's contributions in the project.

Participants: All members

**11. Functional Requirement 01**

Product: Login (FR1)

Summary: Ensures only registered and authenticated users are allowed onto the application.

Participants: All Members

The task is split into activities focusing on fundamental aspects of the requirement and testing.

- 11.1. **FR1A1:** FR1 Login Functionality
- 11.2. **FR1A2:** FR1 Error/Unsuccessful Login Handling
- 11.3. **FR1A3:** FR1 Questionnaire Display
- 11.4. **FR1A4:** Testing for FR1

## 12. Functional Requirement 02

Product: User Persona (FR2)

Summary: this functional requirement classifies users into persona in order to assist the app tailor its offerings according to the user's needs

Participants: All Members

The task is similarly split into activities focusing on fundamental aspects of the requirement and testing.

- 12.1. **FR2A1:** FR2 Persona Classification
- 12.2. **FR2A2:** FR2 Pre-defined Persona Display
- 12.3. **FR2A3:** FR2 User Input Capture Functionality
- 12.4. **FR2A4:** FR2 Store User Input Functionality
- 12.5. **FR2A5:** Testing for FR2

## 13. Functional Requirement 03

Product: Use Case (FR3)

Summary: this functional requirement determines a user's use case

Participants: All Members

The task is similarly split into activities focusing on fundamental aspects of the requirement and testing.

- 13.1. **FR3A1:** FR3 User's Use Case Determination from Answers/From FR2
- 13.2. **FR3A2:** Testing for FR3

## 14. Functional Requirement 04

Product: Personalisation (FR4)

Summary: this functional requirement will display a personalised landing page to the user

Participants: All Members

The task is similarly split into activities focusing on fundamental aspects of the requirement and testing.

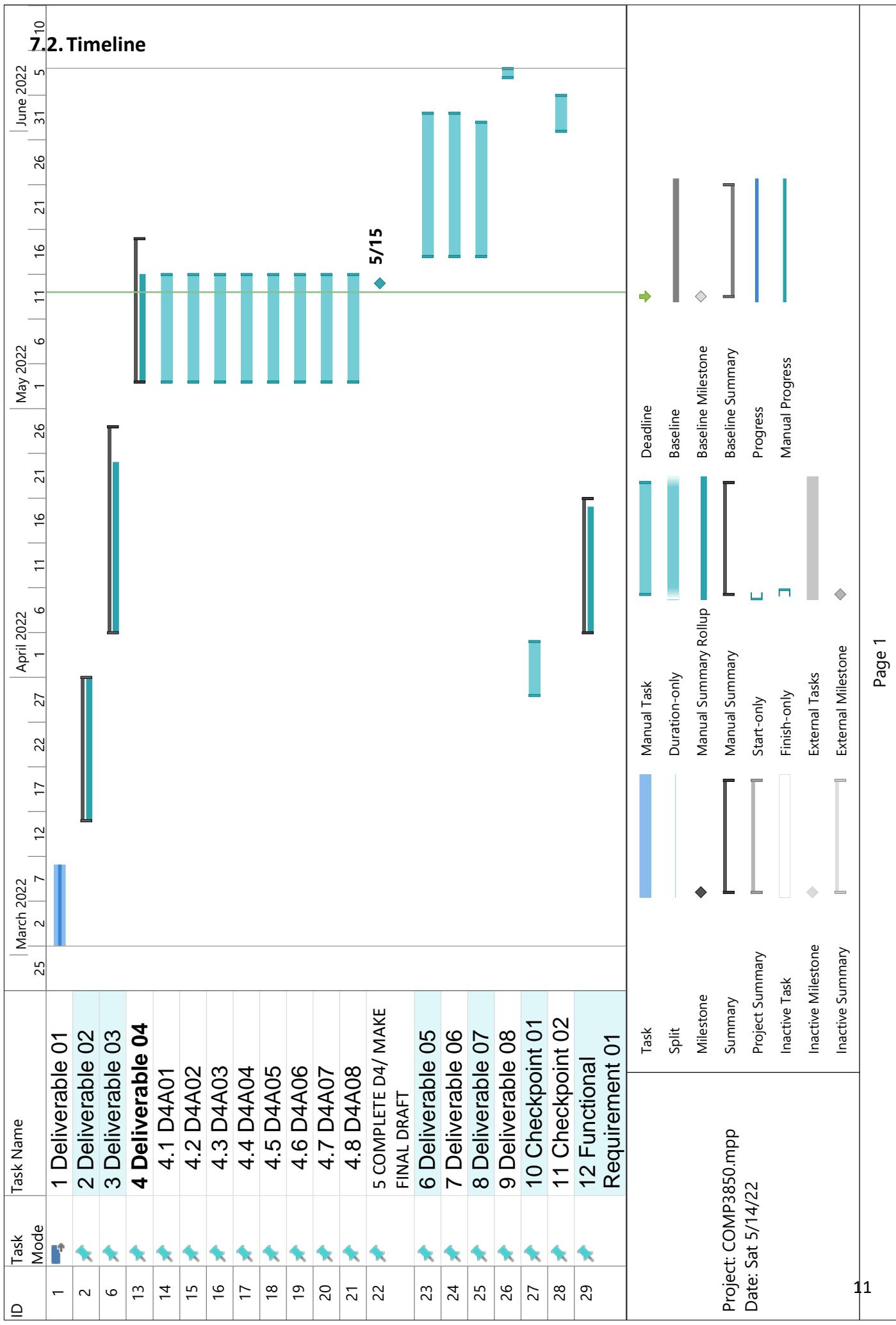
- 14.1. **FR4A1:** FR4 Welcome Functionality
- 14.2. **FR4A2:** FR4 Relevant Features Display
- 14.3. **FR4A3:** Testing for FR4

## 15. Final Testing

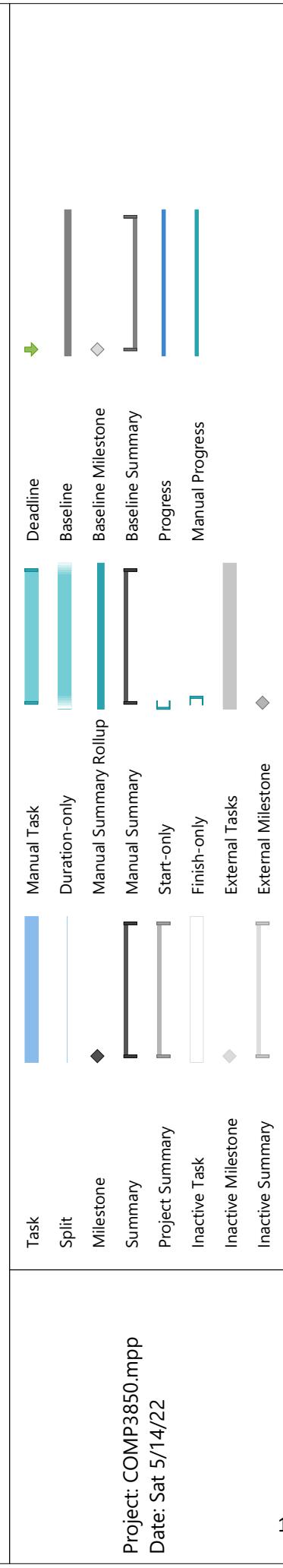
Product: Testing for the Entire Web Application

Summary: after all Functional Requirement are completed and tested, all modules will be integrated and tested to deliver a satisfactory product

Participants: All Members



ID	Task Mode	Task Name	March 2022	April 2022	May 2022	June 2022
			25	2	1	6
			2	12	16	21
34	◆	13 Functional Requirement 02				
40	◆	14 Functional Requirement 03				
43	◆	15 Functional Requirement 04				
47	◆	16 Product Testing				



### **7.3. Resources Allocated**

The resources we have been allocated for this project can be categorised as personnel, hardware and software. The personnel resources that we have been allocated consists of all the individuals that are assigned to the project. This includes each of the group members that are working on the project, as well as any technical skills that each group member has, and the representatives that have been sent by Faethm to give us feedback and advice on the project. The hardware resources we have is any hardware resources that is owned by the group, i.e. laptops/computers and any hardware resources that we have permission to use from Macquarie University or Faethm. Since we are developing an online platform, we have mostly been allocated software resources by Faethm for development. Faethm has given us their website and their component library to use for development, and a Figma document where they have compiled their UI conventions as well as the personas for the user profiles that we need to categorise. Faethm has also compiled resources for us containing the basics of the software that we will be using to develop the project in (JavaScript, Typescript, ReactJS, NodeJS). We have also been provided with some of Faethm's insights, where they break down all the features of their platform and some diagrams explaining the workflow of the processes that are currently in place and have been planned.

All our code for the project is stored in our private GitHub repository, while all documentation is stored in a shared Google Drive folder. Any diagrams required are drawn using Lucid Chart. Documentation for this project is primarily done in Google Docs. The team has collectively agreed to use Discord as our main communication channel while we use Slack and Zoom for communicating without sponsors. We mainly use VSCode as an editor for projects.

## **8. Process Model**

The process model that we are using for this project is agile project management. The agile method allows for Faethm to maintain more control over the project because they are more easily able to make changes to the software requirements. It will also allow them to provide input at regular intervals since the work is completed in a continuous stream and not all at once. This means that they will be able to provide feedback on completed work and request changes or additional features if necessary. The agile method is the ideal method for this project because it has the lowest chance of having the project fail due to the way that work is constantly being done over the development period, which as a result will make the progress of the project predictable. The software will continuously be improved during the entire development period with features and changes being made at the request of Faethm, which allows the project to maintain flexibility on the work that needs to be done and the features that need to be implemented for the project to be successful. This agile method also suits the project because the work that needs to be done is not confirmed and may be easily changed so any work that is done can be modified or built upon as development continues.

## 9. Documentation

The documentation required for this project include:

1. **Feasibility Study:** Discusses the viability of the project, what the nature of the problem is and what needs to be done to address the current situations. Possible solutions for the problem are addressed here and we determine what we need to do to make the project successful.
2. **Project Plan:** Addresses the organisation of the project i.e. Potential risks that could potentially prevent the project from succeeding, the tasks that need to be completed for the project to be successful and how the projects resources will be allocated to ensure the project runs smoothly. A timeline is also given to outline the progression of development. This will need to be updated if the requirements of the project change.
3. **Project Requirements & Scoping Document:** Goes over the purpose of the software and an overview of how it will function. This includes documentation of how the software will work as well as an overview of all of the product's functions. The design restrictions and constraints are also discussed in this document. The functional requirements can be updated throughout development if the need to change the project requirements comes up.
4. **Analysis Document:** Contains a case diagram showing the actors and use cases and the relationships between them. There should be a case diagram for each piece of functionality in the system.
5. **Design Document:** Contains the basic architecture of the system, explaining how processes in the system will be handled as well as explaining the design choices with justifications and trade-offs.
6. **Testing Documentation:** Contains plans for testing strategy, testing types, testing schedule, testing tools and resources assigned, testing milestones and test deliverables. This document should cover all the testing processes used.
7. **User Manual:** Contains documentation for users who are unfamiliar with the software. It should allow a moderately computer-literate user to fully utilise the software's functionality. Contents of the document should contain information that will benefit the user such as an installation guide, configuration settings, screenshots with example data, training, troubleshooting help etc.

## 10. Assumptions

This project plan was made under the assumptions that:

- A1. We will be able to maintain contact with the representatives from Faethm for the duration of the project
- A2. Faethm will provide us with the appropriate resources to complete the project.
- A3. All group members have access to appropriate hardware and software to complete the project.
- A4. The project's requirements can be changed by Faethm at any point in the project if it is required.
- A5. The Team has the time, knowledge, resources and guidance to deliver the project.
- A6. The Team has time to complete all required testing as to deliver a functional bug-free application.
- A7. All team members are free of health and other complications that deters their capability to deliver.
- A8. All team member takes required measures as to NOT be infected in the ongoing pandemic
- A9. The client/user accessing the platform has the authorisation and valid credentials.
- A10. The client/user has represented an organisation that has submitted necessary data for Faethm to analyse.



# Scoping Document



Lance Te, Sepehr Torfeh Nejad, Marcus Ikeda, Erik Horvath, Rojwal Shrestha

# Contents

<b>1. Introduction .....</b>	<b>2</b>
1.1    Purpose.....	2
1.2    Scope .....	2
1.2.1.    Data Flow Diagram .....	3
1.2.2.    Logic Flow Diagram.....	3
1.2.3.    Context Diagram.....	3
1.3    Definitions, Acronyms, and Abbreviations .....	4
1.4    References .....	4
1.5    Overview / Document Convention / Intended Audience.....	4
1.5.1.    Overview.....	4
1.5.2.    Intended Audience .....	4
<b>2. Overall Description.....</b>	<b>5</b>
2.1    Product Perspective.....	5
2.2    Product Functions.....	5
2.3    User Classes and Characteristics .....	5
2.3.1.    Faethm Developers.....	5
2.3.2.    Users .....	5
2.4    Operating Environment.....	6
2.5    User Documentation .....	6
2.5.1.    Faethm Developers.....	6
2.5.2.    Users .....	6
<b>3. Requirements.....</b>	<b>7</b>
3.1    Functional Requirements .....	7
3.1.1.    FR1 – Login.....	7
3.1.2.    FR2 – User Persona.....	7
3.1.3.    FR3 – Use Case.....	7
3.1.4.    FR4 – Personalisation .....	7
3.1.5.    FR5 – Landing Page .....	7
3.2    Design and Implementation Requirements / Constraints.....	8
3.2.1.    Time .....	8
3.2.2.    Technical.....	8
3.3    Usability Requirements .....	9
3.3.1.    Accessibility .....	9
3.3.2.    Efficiency.....	9
3.3.3.    Intuitiveness .....	9
3.4    Other Non-functional Requirements.....	10
<b>4. Client Feedback .....</b>	<b>11</b>
4.1    Meeting Date and Time .....	11
4.2    Feedback Received .....	11
4.3    Team Response.....	11

# 1. Introduction

Faethm is a data driven, AI incorporated analytic company which provides a number of services on their platform for their users. They have partnered with Macquarie University to provide industry experience opportunities for students. Their aim is to develop a personalised landing page to provide new users a better onboarding experience when using the Faethm application for the first time.

## 1.1 Purpose

The purpose of this Software Requirements Specification (SRS) is to provide an overview of both the context and functionality of the Onboarding UI for the Faethm application.

This document serves as a medium between stakeholders (Faethm) and developers (Techpad Technology) to discuss and clarify the functionalities of the Onboarding UI to ensure both parties' expectations are met before any development begins. Given the preliminary nature of this document, both parties can make agreed changes throughout.

Once the development phase commences, the SRS document will then serve as a reference for the developer team to follow and stay on track while developing the Onboarding UI.

Thus, the SRS must be a collaborative effort between stakeholders and developers to produce a well-defined, clear and completely understood document.

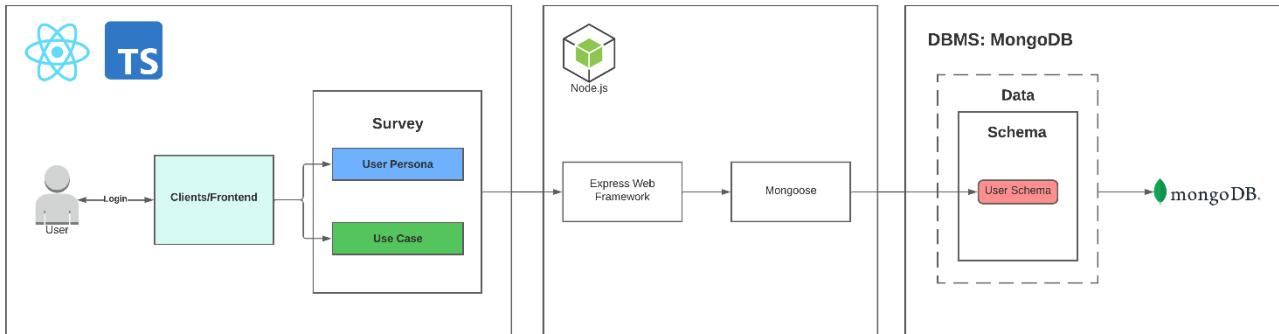
## 1.2 Scope

The Onboarding UI project is an additional data driven step for First Time User (FTU) which provides users with the opportunity to customise their experience on the Faethm application. The UI provides a tailored landing page for users and improves the user experience (UX) on the Faethm platform. The UI is a Single Page Web Application (SPA), with MERN framework (ReactJS, NodeJS, MongoDB) which may later be integrated with the Faethm web application.

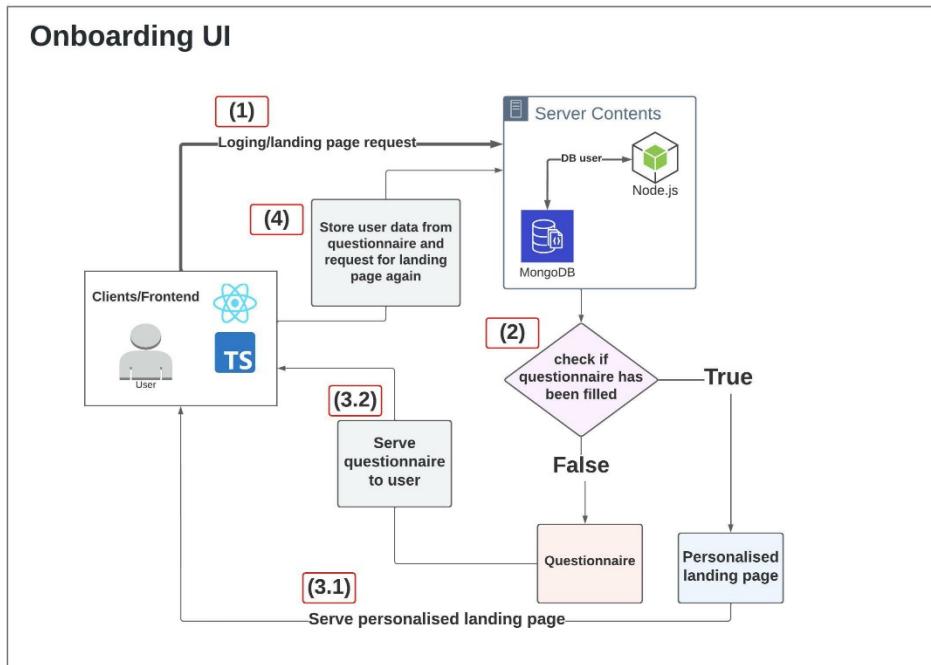
Comprising the first-time user experience (FTUE), the Onboarding UI is only applicable to users logging in for the first time. The UI **will** check if the user has previously logged in. For first time users, the UI serves a short set of questions. This data is stored for future reference and used to assign users a predefined persona and determine the most relevant services available on the platform to them. Ultimately, this enables the UI to provide a personalised landing page. The UI **will not** serve the short set of questions to users that have previously logged in. Instead, their personalised landing page will be served directly.

The Faethm platform offers a number of different services and they seek to enhance the overall user experience. By providing a personalised landing page, the Onboarding UI streamlines the user experience, offers Faethm greater insight into their user, and offers potential avenues for the development of future products.

### 1.2.1. Data Flow Diagram



### 1.2.2. Logic Flow Diagram



### 1.2.3. Context Diagram



### **1.3 Definitions, Acronyms, and Abbreviations**

<b>Acronyms/Abbreviations</b>	<b>Definitions</b>
AI	Artificial Intelligence
API	Application Programming Interface
CSRF	Cross-Site Request Forgery
FTUE	First Time User Experience
SPA	Single Page Web Application
SRS	Software Requirement Specification
UI	User Interface
UX	User Experience

### **1.4 References**

Systems and software engineering – Life cycle processes – Requirements engineering (ISO/IEC/IEEE 291448, 2011). Obtained from:

<https://login.simsrad.net.ocs.mq.edu.au/login?url=https://ieeexplore.ieee.org%2fdocument%2f6146379>

### **1.5 Overview / Document Convention / Intended Audience**

#### **1.5.1. Overview**

In this document, product perspective and functions are described. It also contains user characteristics and the operating environment of the application.

Functional requirements of the UI, design and implementations requirements and usability of the application are included.

#### **1.5.2. Intended Audience**

Stakeholders (Faethm) and developers (Techpad Technology) are the intended audience of this document. It aims to inform the developers and the stakeholders of the purpose of this project and clarifies the requirements of the application for both parties.

## 2. Overall Description

### 2.1 Product Perspective

Our Onboarding UI will serve as the First Time User Experience (FTUE). This will be the first page a user interacts with upon logging into the Faethm application for the first time.

While we develop this feature, the interface will remain **independent** of the rest of the Faethm app. When implemented into the Faethm app, the interface will sit just behind the login screen and provide users with direct access to the tools and articles most relevant to their use case.

### 2.2 Product Functions

Our Onboarding UI will provide users with a personalised dashboard that caters to their needs and what they hope to gain out of the app. The UI offers two core functionalities:

- To determine the persona of a user (refer to *User Classes and Characteristics*)
- To determine a user's use case

By determining these from the user, the UI will provide the appropriate insights (workforce, technology, job) and articles that will align with their use case.

### 2.3 User Classes and Characteristics

There are two user classes that will interact with the Onboarding UI: Faethm developers and users.

#### 2.3.1. Faethm Developers

The interaction of Faethm developers with the Onboarding UI will involve maintaining and updating the UI to meet their user's evolving needs. This could include updating personas to reflect changes in the type of employees in the corporate structure to general maintenance of the UI.

#### 2.3.2. Users

Users will make up the primary group interacting with the Onboarding UI. They will answer a short set of questions the first time they login. This is used to classify users into one of nine broadly generalised personas based on their occupation and position within their company. These personas are:

- |                         |                       |                          |
|-------------------------|-----------------------|--------------------------|
| 1. Concerned executive  | 4. Overbooked manager | 7. Workforce planner     |
| 2. Head of the division | 5. Lost employee      | 8. Generalist HR manager |
| 3. Data loving analyst  | 6. Learning manager   | 9. Automation engineer   |

Each type of user has varying levels of technical expertise, capabilities, and accessibility requirements. This is a strong consideration for the product's design requirements to ensure the UI is intuitive across all user types.

## **2.4 Operating Environment**

Within the context of users, the Onboarding UI will operate through their web browser whether that be on a laptop or mobile device. This requires the UI, as with any modern UI, to be able to adapt to different screen sizes and aspect ratios.

Within the context of the Faethm developers, the Onboarding UI should operate and will thus be built using the MERN (MongoDB, Express, React, Node) stack for ease of integration with the rest of the Faethm application. The different layers of the UI will be developed using technologies across the MERN stack:

- Frontend: React.js
- Server: Express.js, Node.js
- Database: MongoDB

The frontend provides a short set of questions to determine their persona and use case. From there, the UI will display personalised content derived from a user's categorisation from the initial questionnaire. The server will handle requests from the frontend and serve the initial page to the user's browser. The database stores user information which includes their answers to the set of questions and the persona they are classified as.

The frontend will be designed to support all modern web-browsers and mobile devices. All existing browser-defined styles will be reset and built from the ground up to ensure a consistent UI experience across all devices.

## **2.5 User Documentation**

### **2.5.1. Faethm Developers**

Documentation of software features, and a map of the code will be provided to the Faethm developers to ensure a smooth handover. These documents will be in a PDF format.

### **2.5.2. Users**

A help hint / tooltip will provide sufficient information for users to interact with the Onboarding UI.

## **3. Requirements**

### **3.1 Functional Requirements**

#### **3.1.1. FR1 – Login**

The login feature will serve as a barrier to ensure only registered and authenticated users are allowed entry into the Faethm application.

- REQ 1. The system shall display the login form.
- REQ 2. The system shall display an error message on unsuccessful login.
- REQ 3. The system shall display a short set of questions upon successful login.

#### **3.1.2. FR2 – User Persona**

Classifying the user into a persona will assist the Faethm app to customise its offerings to better suit the needs of the user.

- REQ 1. The system shall classify users into one of nine persona classes.
- REQ 2. The system shall provide a user form displaying the nine pre-defined personas with an icon graphic and short description for each.
- REQ 3. The system shall capture user input, and store that information in the database.

#### **3.1.3. FR3 – Use Case**

Determining a user's use case will assist the Faethm app to customise its offerings to cater more specifically to user needs.

- REQ 1. system shall provide a short set of questions each with a set of possible answers.
- REQ 2. The system shall determine the user's use case derived from their answers from these questions.

#### **3.1.4. FR4 – Personalisation**

The system shall use FR1 and FR2 to display a personalised landing page to the user.

- REQ 1. The system shall provide a welcome message customised with the user's name and organisation.
- REQ 2. The system shall provide tools relevant to the user.
- REQ 3. The system shall provide articles relevant to the user.

#### **3.1.5. FR5 – Landing Page**

- REQ 1. The system shall display a landing page to the user in accordance with the style and format of the Faethm application.
- REQ 2. Display a button that triggers FR1.

## **3.2 Design and Implementation Requirements / Constraints**

### **3.2.1. Time**

Currently, the team is balancing part- or full-time work alongside this project. This was an important consideration when proposing a realistic timeframe. Ultimately, it was agreed that a reasonable timeframe to develop the documentation and final deliverable for the Onboarding UI ranges from 13 weeks to a maximum of 16 weeks. A breakdown of the deliverable's timeline is outlined under 'Project Schedule' in the Project Plan.

### **3.2.2. Technical**

There are two technical considerations for the Onboarding UI.

1. The Faethm application adheres to a distinct style throughout which is clearly defined in their CSS. Thus, the Onboarding UI must continue to maintain a **consistent UI**.
2. The Faethm application was developed using the MERN stack. Thus, the Onboarding UI must be developed in the **MERN stack** to ensure integration compatibility.

### **3.3 Usability Requirements**

#### **3.3.1. Accessibility**

The Onboarding UI should:

- REQ 1. Offer a responsive design to serve users a consistent experience whether they are accessing the UI on a mobile device or laptop.
- REQ 2. Be browser-neutral to ensure users are not limited to interacting with the UI on certain web browsers.

**Fit Criteria:** The web application should be accessible on all devices.

#### **3.3.2. Efficiency**

The Onboarding UI should maintain reasonable load times for efficiency, and to improve the user experience.

- REQ 1. The frontend should make minimal requests to the server and have clear logic.
- REQ 2. The server should be well structured and have a well-defined API.

**Fit Criteria:** The web application should be able to light-weight and load within a few milliseconds.

#### **3.3.3. Intuitiveness**

The Onboarding UI should:

- REQ 1. Be user-friendly, that is, easy to learn and navigate. Additional help hints or tooltips may be implemented to increase user-friendliness.
- REQ 2. Provide a low perceived workload to ensure a smooth learning experience while interacting with the UI.

**Fit Criteria:** The web application should be easy to navigate such that users should be able to appreciate its features within five minutes.

## 3.4 Other Non-functional Requirements

### 3.4.1. Performance

REQ 1. Support lower end devices

- Devices accessing the Onboarding UI should be JavaScript supported due to the nature of the software (React).
- The Onboarding UI shall be a relatively lightweight software so it can be accessed by lower end devices.

REQ 2. Support slower internet connections

- The Onboarding UI shall minimise the required API request by implementing clear and well-thought-out frontend logic to ensure a consistently smooth experience despite a slower internet connection.

REQ 3. Support an increasing numbers of simultaneous users

- The Onboarding UI shall be hosted on a server with appropriate infrastructure to support a growing user base.
- Load-balancing and intelligent scheduling algorithm is required during peak time.

REQ 4. Instantaneous load times

- The frontend React components shall be optimised to avoid long initial loading times of the UI.

**Fit Criteria:** The web application should be able to run on older devices. We have reasonably defined older devices as those that are up to 3 generations old.

### 3.4.2. Security

REQ 1. Authentication

- The Onboarding UI shall only allow registered users who provide the correct login credentials into the application.
- The Onboarding UI shall deny access to unregistered or unauthenticated users.

REQ 2. Authorisation / Access

- The Onboarding UI does not distinguish between admin users and general users. However, user access will be restricted to authorised workspaces.
- Users shall only be able to view their workspaces.

REQ 3. CSRF Prevention

- The Onboarding UI shall implement JSON Web Tokens (JWT) to mitigate the risks of Cross-Site Request Forgery (CSRF).

**Fit Criteria:** The web application should only allow authenticated users into the system, and only provide access to authorised parts of the application.

### 3.4.3. Scalability

As the Faethm platform continues to expand, the need for a scalable solution grows with it.

REQ 1. The Onboarding UI shall be developed in a manner that provides a level of convenience for the Faethm developers to expand or shrink the number of questions.

REQ 2. The Onboarding UI shall also be developed to handle a growing number of concurrent users.

**Fit Criteria:** The web application should be able to handle a large number of concurrent users.

## **4. Client Feedback**

### **4.1 Meeting Date and Time**

We submitted our *Scoping Document* for review via email and GitHub on Monday 28 March 2022, 9:00am AEDT. In addition, we submitted our entire *Deliverable 2* to provide context to the *Scoping Document*.

### **4.2 Feedback Received**

The feedback we received was positive and confirmed our vision of the Onboarding UI aligned with those of the Faethm team.

### **4.3 Team Response**

There are no further response / action points required at this stage.



# Analysis Document



Lance Te, Sepehr Torfeh Nejad, Marcus Ikeda, Erik Horvath, Rojwal Shrestha

# Contents

<b>1. Minimum Viable Prototype.....</b>	<b>2</b>
1.1 Login .....	3
1.2 Landing Page.....	4
1.3 Persona Page .....	5
1.4 Survey Page .....	8
1.5 Loading Page.....	11
1.6 Navigation Bar .....	12
1.7 Storage Hosting: MongoDB .....	13
1.8 Software Hosting: GitHub.....	14
<b>2. Client Feedback.....</b>	<b>15</b>
2.1 Week 8 Meeting .....	15
Meeting Date and Time .....	15
Feedback Received .....	15
Team Response.....	16
2.2 Week 9 / 10 Meeting .....	17
Meeting Date and Time .....	17
Feedback Received .....	17
Team Response.....	17
2.3 Week 11 Meeting .....	18
Meeting Date and Time .....	23
Feedback Received .....	23
Team Response.....	22
<b>3. Use Case Diagram .....</b>	<b>24</b>
<b>4. Use Case Descriptions and User Stories .....</b>	<b>25</b>
4.1 Login .....	25
4.2 Landing Page.....	26
4.3 Persona Page .....	27
4.4 Survey Page .....	28
4.5 Personalised Dashboard.....	29
<b>Assumptions.....</b>	<b>30</b>

## 1. Minimum Viable Prototype

In the time since our demonstration of the Onboarding UI to the client, Faethm AI, we have continued to build on our minimum viable product (MVP). The feedback provided during the demonstration served as a focal point for future iterations of the MVP. In response to this feedback, the team has modified the Onboarding UI to reflect the features and design elements the client desires.

To highlight the changes and / or new features implemented into the Onboarding UI, we use the key words “Feedback” and “Action” throughout this document.

Kindly note, the screenshots below is the actual output of our code (except the dashboard) which can be accessed via: <https://github.com/LanceWhitehorn/Faethm/>

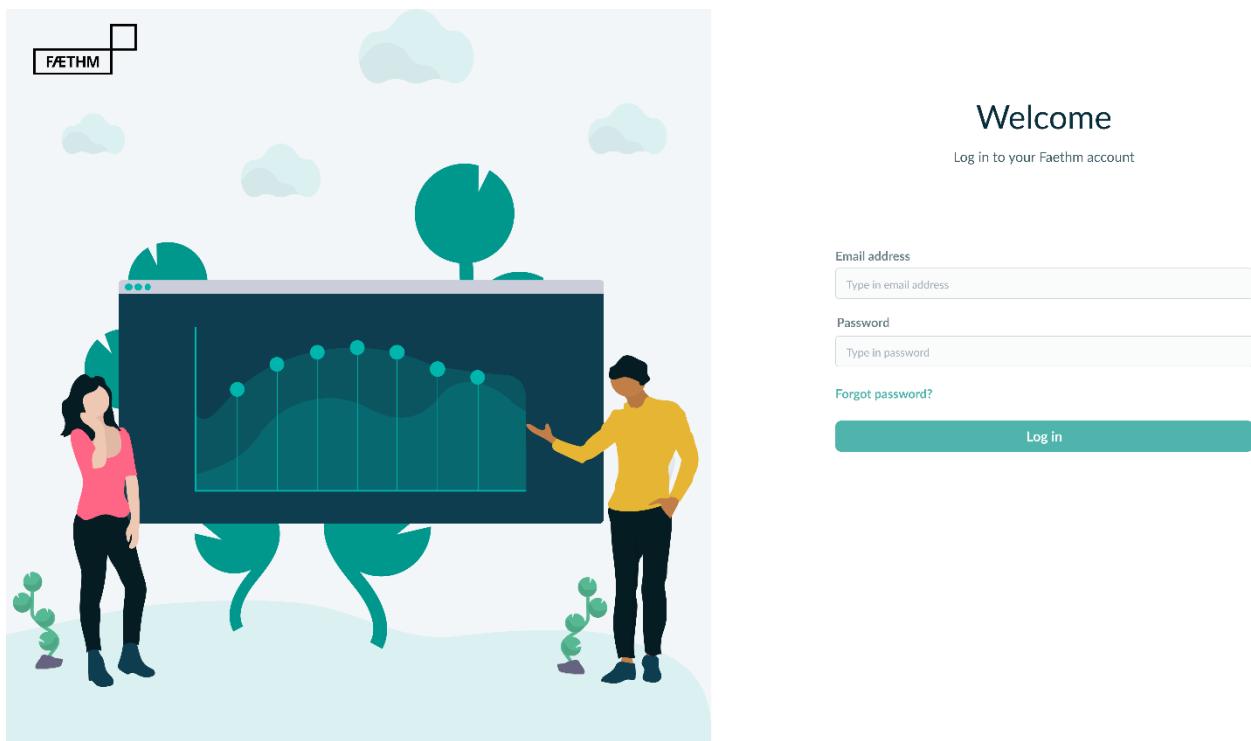
Also, before developing any code, we designed the UI on Figma. This can be accessed via:  
<https://www.figma.com/proto/bCtAFQNPHas1c61rER3t0Z/Onboarding-UI?node-id=0%3A1>

## 1.1 Login

In order to access any part of the Faethm AI application, including our Onboarding UI, a user will need to be authenticated. This is achieved through the login form which will allow the user to verify their identity. The login page is patterned off Faethm's existing login page to ensure a consistent user interface.

To login, users will need to provide two things: the email address and password used when they registered for their Faethm account. The email address input field will have validation rules that ensure that a valid email address is provided that includes the username, an @ symbol, domain name, a dot, and the domain. The email address and password will need to match those on record for a user to access the rest of the Faethm application.

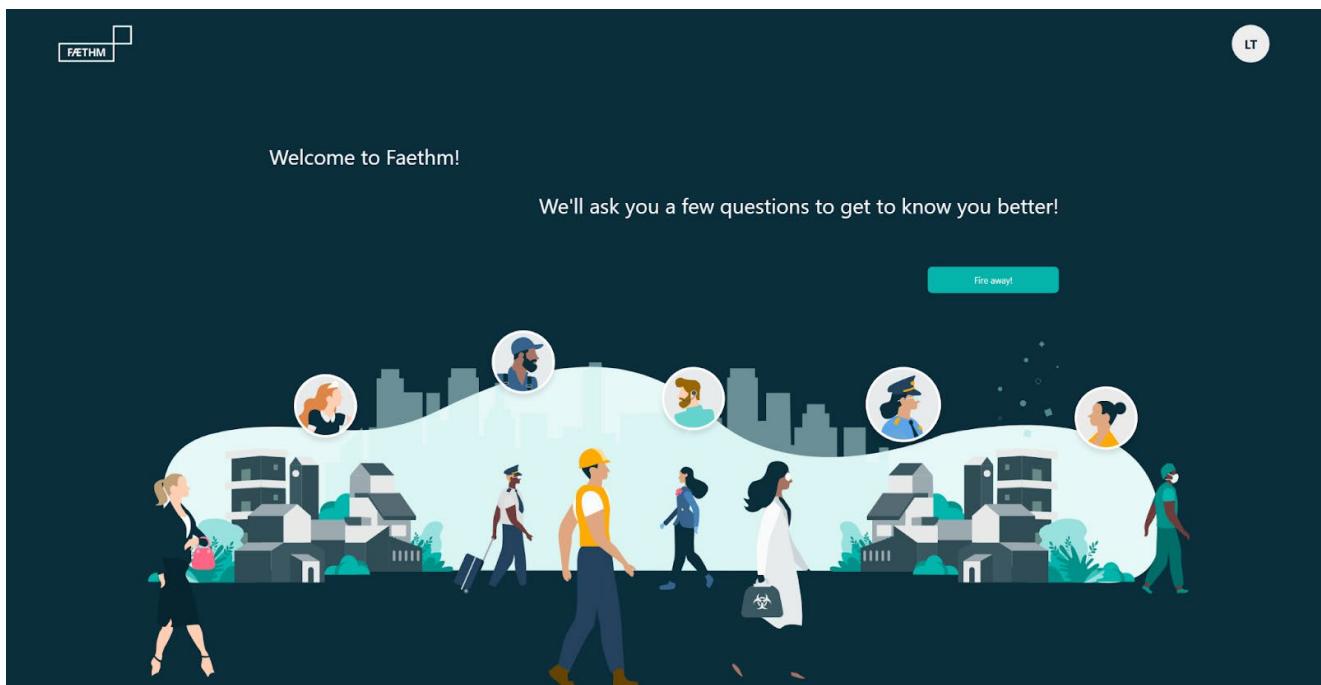
Once a user has provided a valid email address and password, and is authenticated as an existing user in the database, a JSON Web Token (JWT) is stored in both the database and the user's local storage. This is to allow constant checking to ensure that the user's session is still valid while they are using the Faethm application.



## 1.2 Landing Page

**Feedback:** The client requested, if possible, for a landing page.

**Action:** We developed a landing page for users that login for the first time. This full-page feature is a simple, yet effective visual element that serves to ease users into the onboarding process and the Onboarding UI.



### 1.3 Persona Page

If a user is logging in for the first time, they will be presented with the Onboarding UI. The Onboarding UI includes a questionnaire that determines the category of employee a user falls under. The possible categories of employees are:

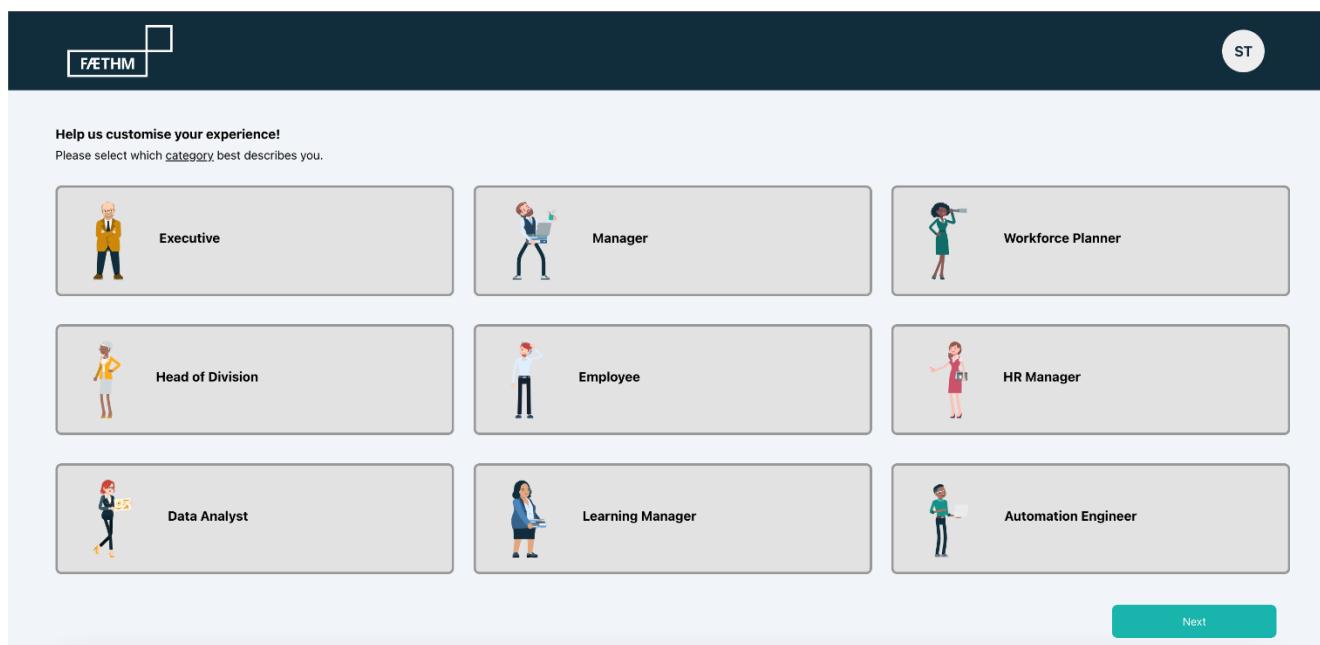
- Executive
- Head of Division
- Data Analyst
- Manager
- Employee
- Learning Manager
- Workforce Planner
- HR Manager
- Automation Engineer

**Feedback:** The client requested that the Persona interface so the user can intuitively learn that an interaction is required. They suggested a card-flip animation.

**Action:** We implemented a card-flipping animation that informs the user that a selection is required from them without the need for additional text on the page. Moreover, moving the description to the 'back' of the cards decreases the amount of text on the screen thereby increasing user readability.

The screenshots below display the four key features / states of the Persona interface:

1. **Initial state:** This is the state which a user logging in for the first time will experience. The large tiles displaying each Persona initially grab the attention of the user. The text above either serves as an explanation of what to do, or a clarification of what the user already intuitively thinks is required.



2. **Hover State:** When a user hovers over the Persona, they are provided with a description of the Persona.

Help us customise your experience!  
Please select which category best describes you.

Executive	A manager ensures that their assigned department, store, or district is well staffed and provisioned, adheres to quality and service standards, increases revenue and market share, and helps the business accomplish its goals.	Workforce Planner
Head of Division	Employee	HR Manager
Data Analyst	Learning Manager	Automation Engineer

Next

3. **Select State:** When a user selects a Persona, the card gains both a darker background and border to provide user feedback of their selection.

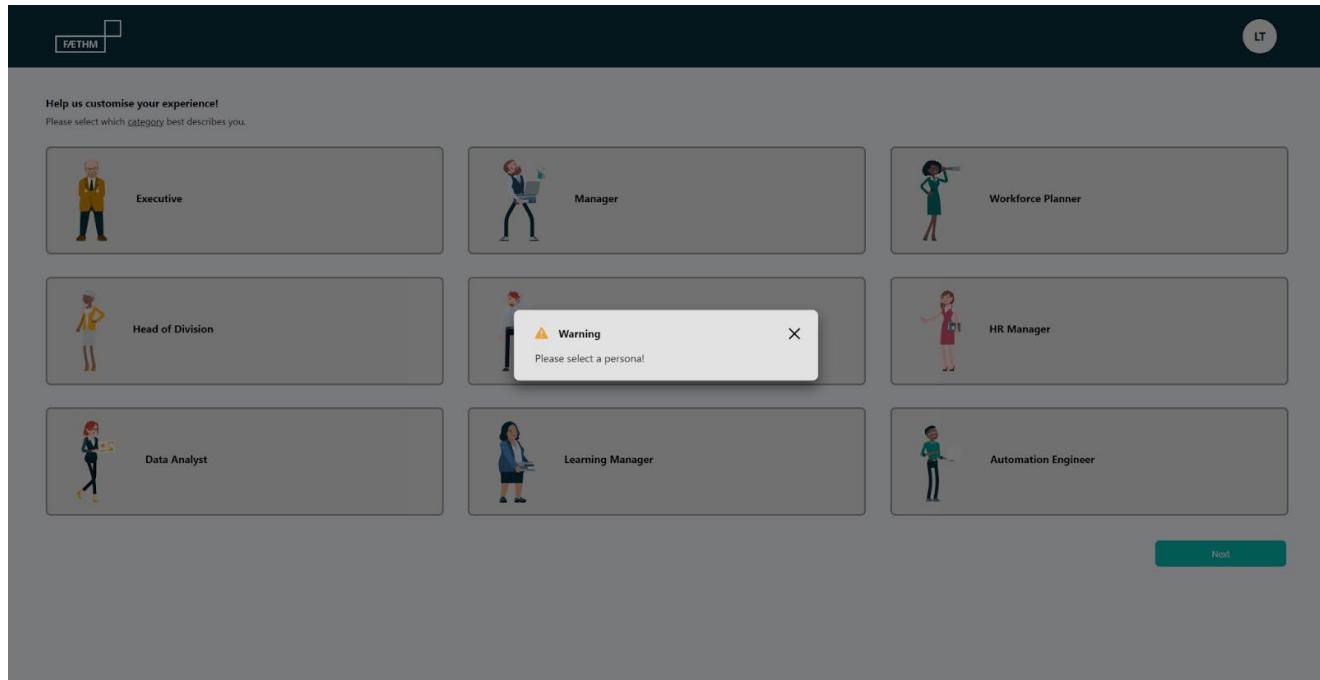
Help us customise your experience!  
Please select which category best describes you.

Executive	Manager	Workforce Planner
Head of Division	Employee	HR Manager
Data Analyst	Learning Manager	Automation Engineer

Next

4. **Validation State:** When no Persona is selected, the user is provided a modal that informs them a selection is required to proceed. To enhance user-friendliness, the modal can be hidden / closed in a few ways:

- Pressing the Close icon in the top right corner
- Pressing anywhere outside the modal
- Pressing the 'Escape' key



## 1.4 Survey Page

In addition to categorising the user, further questions are asked to determine the user's use-case of the Faethm application to better understand and cater to their needs and requirements. While the exact questions have not been finalised, these are some that may be used in production:

1. To what extent do you plan to maintain and upskill your current workforce?
2. To what extent is your workforce willing to learn new skills and undertake further studies?
3. To what extent do you believe your industry will be impacted by emerging technologies?
4. To what extent do you think your industry will be impacted by automation?

Ultimately, this data is used to create a personalised dashboard for the user as part of the Onboarding UI.

**Feedback:** The client requested the team meet their Client Insights team to design better questions to extract maximum insight from the user.

**Action:** After discussions with the Client Insights team, we modified the Onboarding UI survey to better capture and understand the user's use-case of the Faethm application.

The screenshots below display the four key features / states of the Survey interface:

1. **Initial state:** This is the state which a user logging in for the first time will experience.

The screenshot shows the initial state of the Faethm Survey interface. At the top, there is a dark header bar with the 'FAETHM' logo on the left and a 'LT' icon on the right. Below the header, there are four survey questions, each with three rating options: Low, Medium, and High. The questions are:

1. To what extent do you plan to maintain and upskill your current workforce?
2. To what extent is your workforce willing to learn new skills and undertake further studies?
3. To what extent do you believe your industry will be impacted by emerging technologies?
4. To what extent do you think your industry will be impacted by automation?

Each question has three input fields: 'Low' (left), 'Medium' (center), and 'High' (right). A large teal 'Submit' button is located at the bottom right of the form area.

2. **Hover State:** When a user hovers over an option, the option is highlighted grey. This provides feedback to the user informing them of their current cursor position over the highlighted option.

This screenshot shows a survey interface with four questions. Each question has three options: Low, Medium, and High. The 'Medium' option for each question is currently highlighted with a grey background, indicating it is the active or selected state. The survey is titled 'FETHM' and includes a 'Submit' button at the bottom right.

1. To what extent do you plan to maintain and upskill your current workforce?

2. To what extent is your workforce willing to learn new skills and undertake further studies?

3. To what extent do you believe your industry will be impacted by emerging technologies?

4. To what extent do you think your industry will be impacted by automation?

Submit

3. **Select State:** When a user selects an option, the option gains both a darker background and border to provide user feedback of their selection.

This screenshot shows the same survey interface as the previous one, but with a different selection. The 'Medium' option for the second question ('To what extent is your workforce willing to learn new skills and undertake further studies?') is now highlighted with a darker grey background and a thicker border, indicating it is the selected state. The other options ('Low' and 'High') are in their original white state. The survey is titled 'FETHM' and includes a 'Submit' button at the bottom right.

1. To what extent do you plan to maintain and upskill your current workforce?

2. To what extent is your workforce willing to learn new skills and undertake further studies?

3. To what extent do you believe your industry will be impacted by emerging technologies?

4. To what extent do you think your industry will be impacted by automation?

Submit

4. **Validation State:** When any option is missing, the user is provided a modal that informs them all selections are required to proceed. To enhance user-friendliness, the modal can be hidden / closed in a few ways:

- Pressing the Close icon in the top right corner
- Pressing anywhere outside the modal
- Pressing the 'Escape' key

The screenshot shows a survey interface with four questions and a warning modal.

1. To what extent do you plan to maintain and upskill your current workforce?

2. To what extent is your workforce willing to learn new skills and undertake further studies?

3. To what extent do you believe your industry will be impacted by emerging technologies?

4. To what extent do you think your industry will be impacted by automation?

A warning modal is displayed over the fourth question's slider. The modal has a yellow warning icon, the word "Warning" in bold, and the text "Your answers will help, we promise!"

At the bottom right, there is a teal "Submit" button.

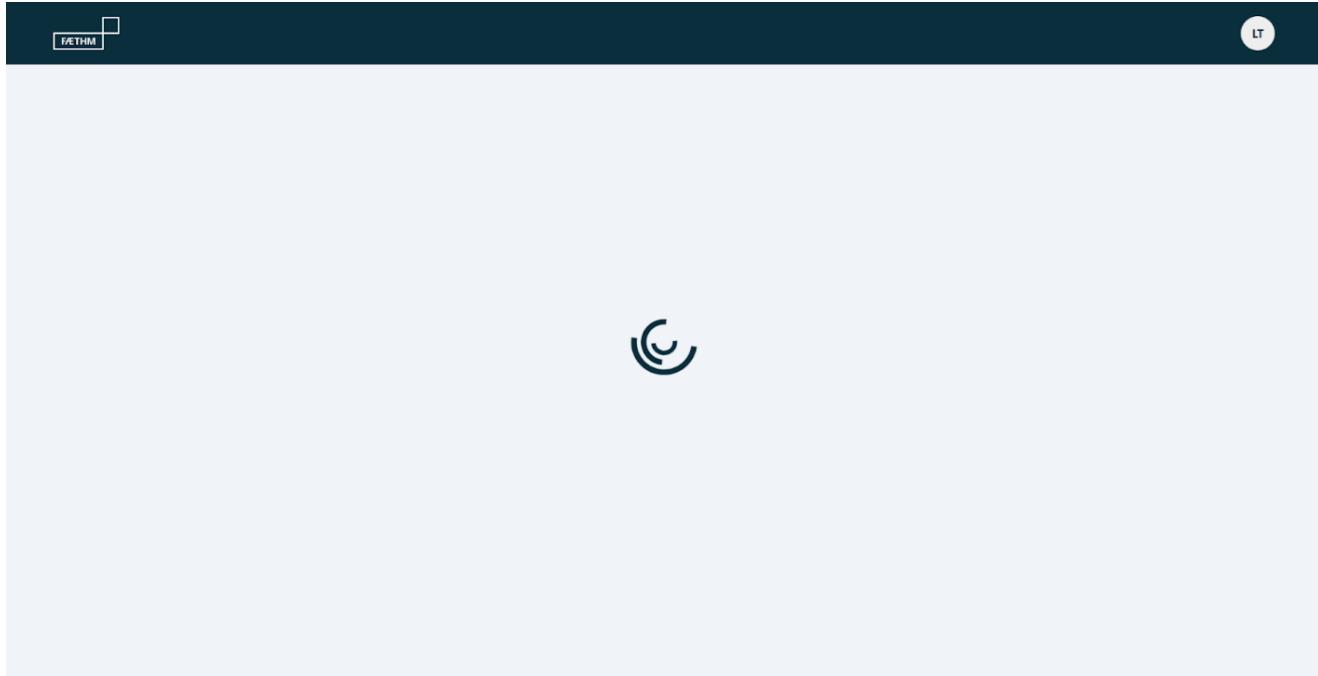
## 1.5 Loading Page

Once the user answers the questionnaire, their data is used to provide a personalised dashboard. Before the dashboard is provided to the user, an intermediate loading page is presented.

While the information being processed from the questionnaire is not large in size, as part of our design choice, the Onboarding UI will display a revolving loading icon. This delay is purposefully built-in to create anticipation and give the perception of added value to the user.

**Feedback:** The client provided positive feedback for a loading page.

**Action:** We implemented a loading page into the MVP.



## 1.6 Navigation Bar

Throughout the Onboarding UI experience, the Navigation Bar only shows the Faethm logo (top left) and the would-be user logo (top right). The links to the other features of the application are not visible or accessible to a first-time user. This ensures that the user proceeds through the Onboarding UI before accessing the rest of the Faethm platform.

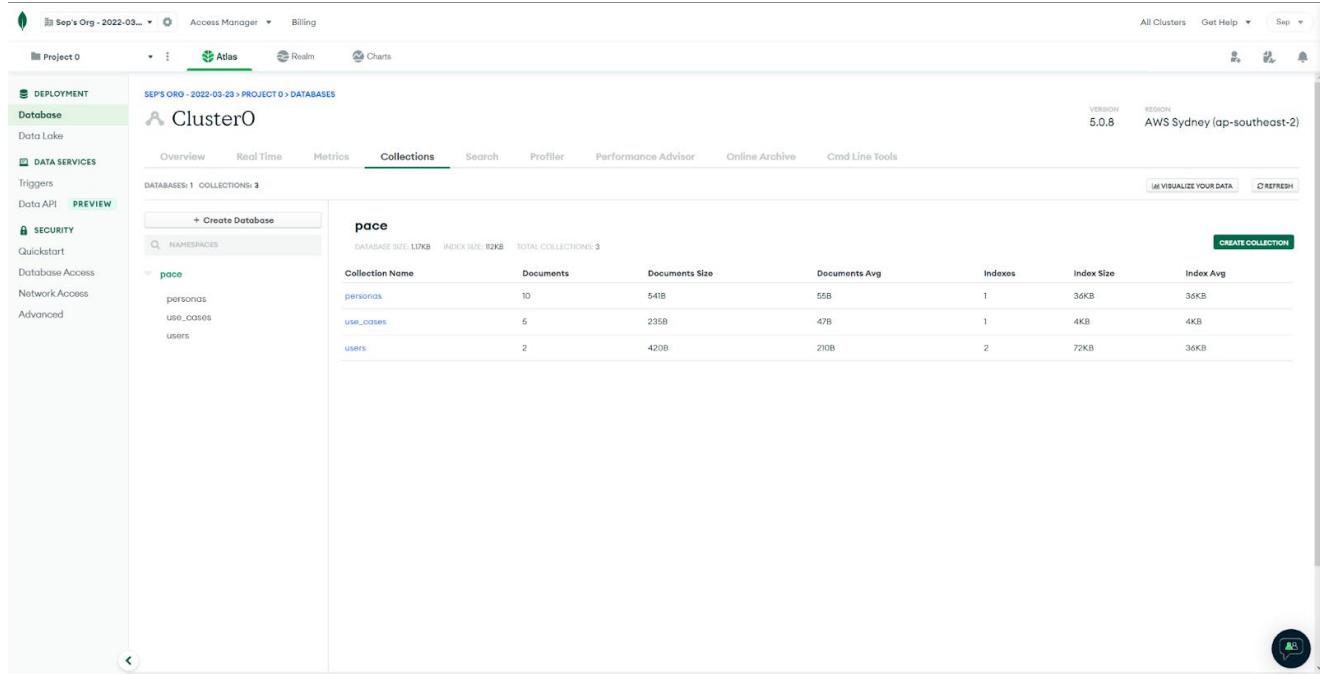
A logout feature has been added should users desire to complete the onboarding process at a later date.



## 1.7 Storage Hosting: MongoDB

The storage solution implemented for the Onboarding UI is Mongo Atlas. The remote MongoDB server stores the Personas, Use Cases, and Users collections. The screenshots below show our Mongo Atlas dashboard.

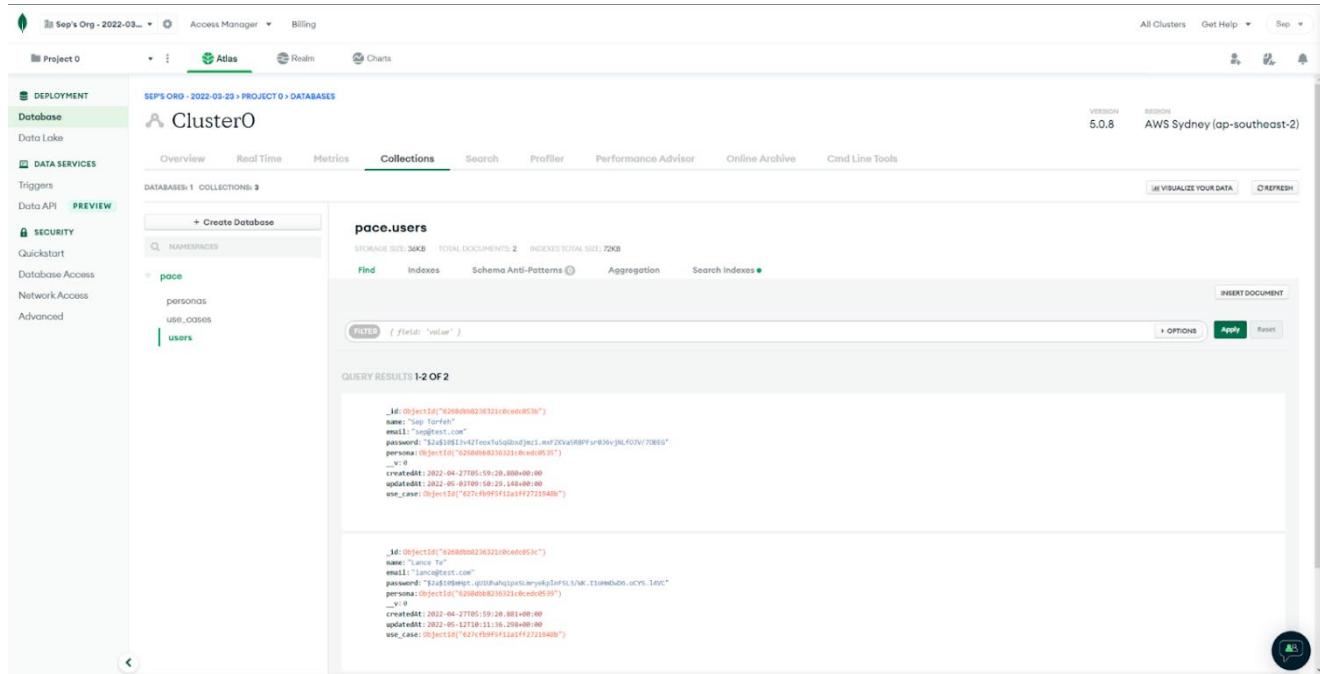
1. There are three collections, that is, Personas, Use Cases, and Users. Both the Personas and Use Cases are relational to Users.



The screenshot shows the MongoDB Atlas interface for the 'Cluster0' database. The left sidebar includes sections for Deployment, Data Services, and Security. The main area displays the 'Collections' tab for the 'pace' database, which contains three collections: 'personas', 'use\_cases', and 'users'. A detailed table provides statistics for each collection, such as document count, size, and average index size.

Collection Name	Documents	Documents Size	Documents Avg	Indexes	Index Size	Index Avg
personas	10	541B	55B	1	36KB	36KB
use_cases	5	235B	47B	1	4KB	4KB
users	2	420B	210B	2	72KB	36KB

2. The Users collection involves both the Personas and Use Case collection. That is, a user will be assigned a persona and a use case which are defined in their respective collections and linked via their respective object IDs.



The screenshot shows the MongoDB Atlas interface for the 'pace' database, specifically the 'users' collection. The left sidebar includes sections for Deployment, Data Services, and Security. The main area displays the 'Collections' tab for the 'pace' database, which contains three collections: 'personas', 'use\_cases', and 'users'. The 'users' collection is selected, showing two documents. The first document is for 'Sep' (ObjectID: 620b0000020321c0edc053b) and the second for 'Lance' (ObjectID: 620b0000020321c0edc053c). Each document includes fields like name, email, password, persona, and use\_case.

```
_id: ObjectId("620b0000020321c0edc053b")
name: "Sep"
email: "septorfeh.com"
password: "12a510513e42f0ef01059d0e1.msf20va58899sr436v3nLFG7v/7086j"
persona: ObjectId("620b0000020321c0edc053b")
use_case: ObjectId("627cf0ff9f11a1ff2721940b")

_id: ObjectId("620b0000020321c0edc053c")
name: "Lance"
email: "lance@lance.com"
password: "12a510513e42f0ef01059d0e1.msf20va58899sr436v3nLFG7v/7086j"
persona: ObjectId("620b0000020321c0edc053b")
use_case: ObjectId("627cf0ff9f11a1ff2721940b")
```

## 1.8 Software Hosting: GitHub

Our code base is hosted on a GitHub repository (<https://github.com/LanceWhitehorn/Faethm/>) that is shared across both the Onboarding UI team and the Faethm AI team. The screenshot below shows the root folder of our code which is structured into a client and server directory.

The screenshot shows the GitHub repository page for 'LanceWhitehorn / Faethm'. The main navigation bar includes 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. The 'Code' tab is selected, showing a 'dev' branch with 2 branches and 0 tags. A message indicates the 'dev' branch is 24 commits ahead of 'main'. Below this, a commit from 'SepTorfeh' is listed: 'cleaning up login state in redux after logout' (commit c1b516a, 18 hours ago). The repository structure on the left shows 'app', 'deliverables', '.gitignore', and 'README.md'. The right sidebar contains sections for 'About' (COMP3850 | Sem 1, 2022), 'Releases' (none), 'Packages' (none), 'Contributors' (SepTorfeh, LanceWhitehorn), and 'Languages' (TypeScript 64.2%, JavaScript 22.1%, CSS 8.7%, HTML 3.7%, Dockerfile 0.3%). A large preview window on the left displays the 'Faethm AI' section of the README, which includes a 'Summary' and a 'The Team' table:

Name	Student ID
Lance Te	45581819
Sepehr Torfeh Nejad	46042547
Rojival Shrestha	46137572

The screenshot below shows the two primary branches we have used throughout this project, that is, *main* and *dev*. While implementing other features, we temporarily worked on separate branches which we merged into the *dev* branch and then ultimately into the *main* branch.

The screenshot shows the GitHub repository branches page for 'LanceWhitehorn / Faethm'. The top navigation bar includes 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. The 'Code' tab is selected, showing the 'Overview' tab is active. The page lists three branches: 'Default branch' (main), 'Your branches' (dev), and 'Active branches' (dev). Each branch entry shows the last update (15 days ago) and the user who updated it (Lance or SepTorfeh). A search bar at the top right allows searching for branches. The bottom of the page includes standard GitHub footer links: © 2022 GitHub, Inc., Terms, Privacy, Security, Status, Docs, Contact GitHub, Pricing, API, Training, Blog, and About.

## 2. Client Feedback

### 2.1 Week 8 Meeting

#### Meeting Date and Time

We submitted our Figma MVP for review via Slack on Tuesday 26 April 2022, 3:00pm AEST and our code for review on Wednesday 27 April 2022. In addition, we will provide a live demonstration for the Faethm team in our weekly Wednesday meeting on 27 April 2022 from 4:00pm.

#### Feedback Received

In terms of functionality and general design, the Faethm team informed us that their expectations had been exceeded. One standout design choice was the purposefully built-in delay via a loading page to create anticipation and give the perception of added value to the user.

However, there was feedback in terms of more specific aspects of the design:

- Card background should be a grey from their Figma file.
- Less text on the persona cards perhaps limiting it to one sentence.
- Rounded elements for cards to make it easier on the eyes.
- Clearer instruction: singular or multiple selection.
- Utilise a larger contrast of font-sizes.
- Card-flip animation.
- Use the font-family that the Faethm application uses.
- (Optional) landing page prior to the survey.

Overall, this is very encouraging and confirms that our vision of the Onboarding UI aligns with their vision.

## Team Response

The team has taken the feedback on-board and discussed these desired changes in our post-client meeting. These changes will be implemented one at a time starting from the lowest hanging fruit, such as utilising a larger contrast of font sizes, and progressing to the more complex changes, such as a card-flip animation.

All the requests have been addressed, which include:

- Using the same font-family that the Faethm application uses.
- Using one of the shades of grey defined on the Faethm Figma file.
- Using a larger contrast of font-sizes.
- Using rounded elements.
- Adding a landing page.
- Implementing a card-flip animation.

The screenshot of our GitHub commits shows these changes have been implemented on our dev branch.

The screenshot displays two sections of GitHub commit history for the 'dev' branch. The first section, dated April 28, 2022, contains six commits by LanceWhitehorn:

- validation check on questions (commit e83e93a)
- removed eslint disable lines (commit 9fbcb0b)
- added loading page (commit ac968f6)
- re-applied login routing for unauth user (commit d686f6f)
- added questions, refactored css (commit 6c20f84)
- added deliverable 3 (commit 340e149)

The second section, dated April 27, 2022, contains five commits:

- added card flip, inherited font-family (commit 5ca1840)
- Merge branch 'dev' of https://github.com/LanceWhitehorn/Faethm into dev (commit 43b8e0b)
- added landing page (commit 3e74555)
- Merge branch 'MVP' into dev (commit 6d573b3)
- refactored modal (commit fc54699)

These changes will be presented in next week's client meeting and, if approved, will be merged to the main branch.

## **2.2 Week 9 Meeting**

### **Meeting Date and Time**

We presented our second live demonstration of our final deliverable in Week 9 of our weekly meetings on Wednesday 4 May 2022, 4:00pm to Mikhail Thornhill and Karen Uli from the Faethm team.

We also presented Deliverable 3 and discussed next steps with Deliverable 4.

### **Feedback Received**

After our initial demonstration of the Onboarding UI, Faethm provided us with feedback regarding the visual and design elements of the UI courtesy of Karen Uli, their Product Designer. These requests and suggestions have been implemented and reviewed by Karen in Week 9 of our weekly meetings (Wednesday, 4 May 2022) as part of the team's weekly meeting with the Faethm team. The feedback we received was very positive with Karen expressing her appreciation of the team's ability to follow-through on her design requests.

### **Team Response**

With the design concerns addressed, the team will continue refactoring our code to ensure its readability and interpretability, and continue work on Deliverable 4 until the next meeting. Unfortunately, Shu from the Faethm team was not available during this meeting to provide feedback on our code. We were informed that she would be back next week.

## 2.3 Week 10 Meeting

### Meeting Date and Time

We presented our third live demonstration of our final deliverable in Week 10 of our weekly meetings on Wednesday 11 May 2022, 4:00pm to Mikhail Thornhill and Shu Tu from the Faethm team.

We also presented the mark we received for Deliverable 3 and showcased our progress on Deliverable 4.

### Feedback Received

The Faethm team expressed concern believing the mark we received for our deliverable was not an accurate reflection of the quality of work.

Shu, a Full-Stack Engineer on the Faethm team, provided the team with comprehensive code-related feedback in Week 11 of our weekly meetings (Wednesday, 18 May 2022). We genuinely appreciate how comprehensive her feedback was as we believe it reflects and reciprocates the concerted efforts of the Onboarding UI team. Her word-for-word feedback has been added below with the only change being that of formatting.

#### Backend

Overall, the login API and persona survey API look great. Database model design and relation also are clear. Just some areas we can optimize and align with industrial standards. A few suggestions for APIs:

##### 1. Token expiry, validation check and refresh handling

It seems we make token valid for 30 days? But in real world, we usually limit it to 30 mins(1800s) and make user's token expired after that for more security purpose. We check the expiry after decoded on the BE and rejected the request if token gets expired and then FE can have UI to ask users to refresh token and resend request, or you can just make users log out.

Source file: app/server/utils/generateToken.js

##### 2. API Rule: only return necessary response data from corresponding API

It seems we return user's token in response of the survey, which makes me a bit confused cos I think it's not necessary. Maybe we can remove it from persona survey response to avoid causing some risks and also save the response loads.

Source file: app/server/controller/surveyController.js

##### 3. Improvements on Error handling

Better manage error codes and messages correctly to help future debug and avoid some misleading message. E.g.

- If it's an error related to authentication or authorisation, we usually throw 401
- If it's an error related to the request variables, like invalid username/email, we usually throw bad request 404
- If it's an error irrelevant to auth but just the item not found from the database, like persona type, then it's better to throw 404 not found instead of 401

For nodejs, you can use built-in http2 library to get those standard codes and errors, e.g.

`http2.constants.HTTP_STATUS_NOT_FOUND`

`http2.constants.HTTP_STATUS_BAD_REQUEST` and etc

Source files: app/server/controller/\*\*

App/server/middlewares/\*\*

#### 4. Middleware of auth validation

Overall, it looks great to me. Just a small tweak we could have when abstracting token from headers. We can avoid using split(' ') as space isn't explicitly excluded from JWT token generation, so to be safer, we can use replace('Bearer ','') instead

[Source file:app/server/middlewares/authMiddleware.js](#)

#### 5. Environment variables management

Make the best use of environment variables. Usually, we have env vars setup for like "sandbox", "development", "stage" and "production". I noticed in the codes, there are several places with checking for a "test" environment, which adds some issues, e.g. unnecessary branch conditions and duplicate code blocks, also modifying one place might affect others. So, to improve the code quality and make env handling consistent across the app, we can focus on passing those things as envs and remove those "test" env checking, e.g

For Test env, we can pass

- NODE\_ENV = test
- PORT = 3001
- MONGO\_URL = your mongodb uri
- JWT\_SECRET = testing

For dev:

- NODE\_ENV = development
- PORT = 3001
- MONGO\_URL = your mongodb uri
- JWT\_SECRET = XXXXX

[Source files: /app/server/config/db.js](#)

[app/server/middlewares/authMiddleware.js](#)

[app/server/utils/generateToken.js](#)

#### 6. Dependencies management

Better to lock some key libraries versions(mongoose) to avoid breaking changes or push package-lock file to help it. Also, we can move packages like `nodemon` to dev dependencies if they are only used as dev tools

[Source files: /app/package.json](#)

## Frontend

Generally, the react app is well structures and also main functionalities are matching with the requirements. The figma design also looks nice. Just a few suggestions for frontend app:

- 1. Keep consistency when writing components' CSS styles**

Some components styled with “styled-component” while some uses .css with classname.

Usually, we only choose one way and won’t have mixed/hybrid styles as it’s hard to maintain and cause confusion.

- 2. Less relying on third-party ui library and practicing writing reusable components on our own as much as possible**

It seems we use material-ui underlying for some key components, but it’d be better if we write those reusable components on our own so we can also practice and upskill our react and CSS skills. Faethm implements its own ui-library purely by ourselves without third-party ui libraries so it can support more custom and flexible cases align with our design.

- 3. Refine schema to store user name so as to add more convenience and flexibility for the FE**

Maybe we can save user firstname and lastname separately in MongoDB so we don’t have to handle some string formatting to split them.

- 4. Make the most use of redux**

It seems we fetch user info from API to the redux and then they are unused in redux. Usually, we should make the most use of redux if we do save states inside, so better to use redux selectors to get user info from redux instead of every time setting and getting them from localstorage. If you keep using localstorage, probably there’s no need to use redux any more as it shouldn’t be ended up like an API fetching tool rather than managing states. So, except for some necessary cases, saving all user info into localstorage might cause some data leaking issue, so we only use it if really needed. Redux states to manage them are enough.

- 5. Make the most benefits of Typescript**

Could have better use of typescript on the frontend, e.g. avoid using vague types like any, defining data type and interface for functions, components and etc. So, we can better type checking, reduce errors and development experience using this language.

## **Full-stack E2E**

### **Testing**

It'd be better if we can cover at least some unit testing for significant functionalities, e.g. some utils, redux reducers and other places we handling key logic.

### **Docker containers[optional]**

By using docker contains for local development and production deployment, we won't have issues like we setup env in local machine, as docker will provide the same environment inside the container it created by the Docker file. So, developers and remote machine won't worry about issues like node, npm version mismatch or machine arch different causing package installation failing and etc.

Usually, we would have Docker file for frontend and a separate one for backend, cos they should be running in different docker containers. Then we have docker-compose.yml file to run client container and server container at the same time and exposed to separate ports and connect to each other.

You can follow this doc: <https://medium.com/bb-tutorials-and-thoughts/react-localdevelopment-with-docker-compose-5a247710f997>

## Team Response

While Shu has stressed that the feedback she provided are only suggestions, the team has nonetheless been encouraged with the amount of effort she has given and is eager to put as many of them into action. As with previous requests and suggestions, these changes have been implemented in serial order from the lowest hanging fruit, such as refactoring styled components into plain-vanilla CSS for consistency, and progressing to the more complex changes, such as utilising a Docker container.

All “Frontend” and “Backend” issues outlined have been addressed, which include:

- Setting the appropriate token expiry.
- Restricting API responses to only provide specifically relevant information.
- Using the industry standard error handling codes.
- Using substring instead of split to retrieve the JWT.
- Adding nodemon to dev dependencies.
- Adding package-lock file to avoid version changes.
- Refactoring styled components into plain-vanilla CSS.
- Splitting first name and last name on the User schema.
- Passing and retrieving a user’s first name and last name from Redux instead of local storage.
- Changing any types to allow better type checking.

The screenshot of our GitHub commits shows these changes have been implemented on our dev branch.

The screenshot shows the GitHub commit history for the 'dev' branch. The commits are organized by date: May 18, 2022; May 17, 2022; and May 16, 2022. Each commit includes the author (SepTorfeh), a brief description, the commit hash, and copy/paste/share buttons.

- May 18, 2022:**
  - added use\_case to db (commit hash: 3580cda)
- May 17, 2022:**
  - cleaning up login state in redux after logout (commit hash: c1b516a)
  - using redux to show initials on navbar (commit hash: e3a0312)
  - added user first name and last name (commit hash: af17185)
  - added nodemon to dev dependencies (commit hash: 427510a)
  - adding package-lock file to avoid version changes (commit hash: e4f8e3d)
  - use substring to get jwt token (commit hash: d527ae2)
  - fixed error codes (commit hash: d24231d)
- May 16, 2022:**
  - only return necessary response data after survey submit (commit hash: d35599c)
  - set token expiry to 30 mins (commit hash: aa60aa9)

These changes will be presented in next week’s client meeting and, if approved, will be merged to the main branch.

## **2.4 Week 11 Meeting**

### **Meeting Date and Time**

We presented our fourth live demonstration of our final deliverable in Week 11 of our weekly meetings on Wednesday 18 May 2022, 4:00pm to Mikhail Thornhill, Shu Tu, and Karen Uli from the Faethm team.

We also presented Deliverable 4.

### **Feedback Received**

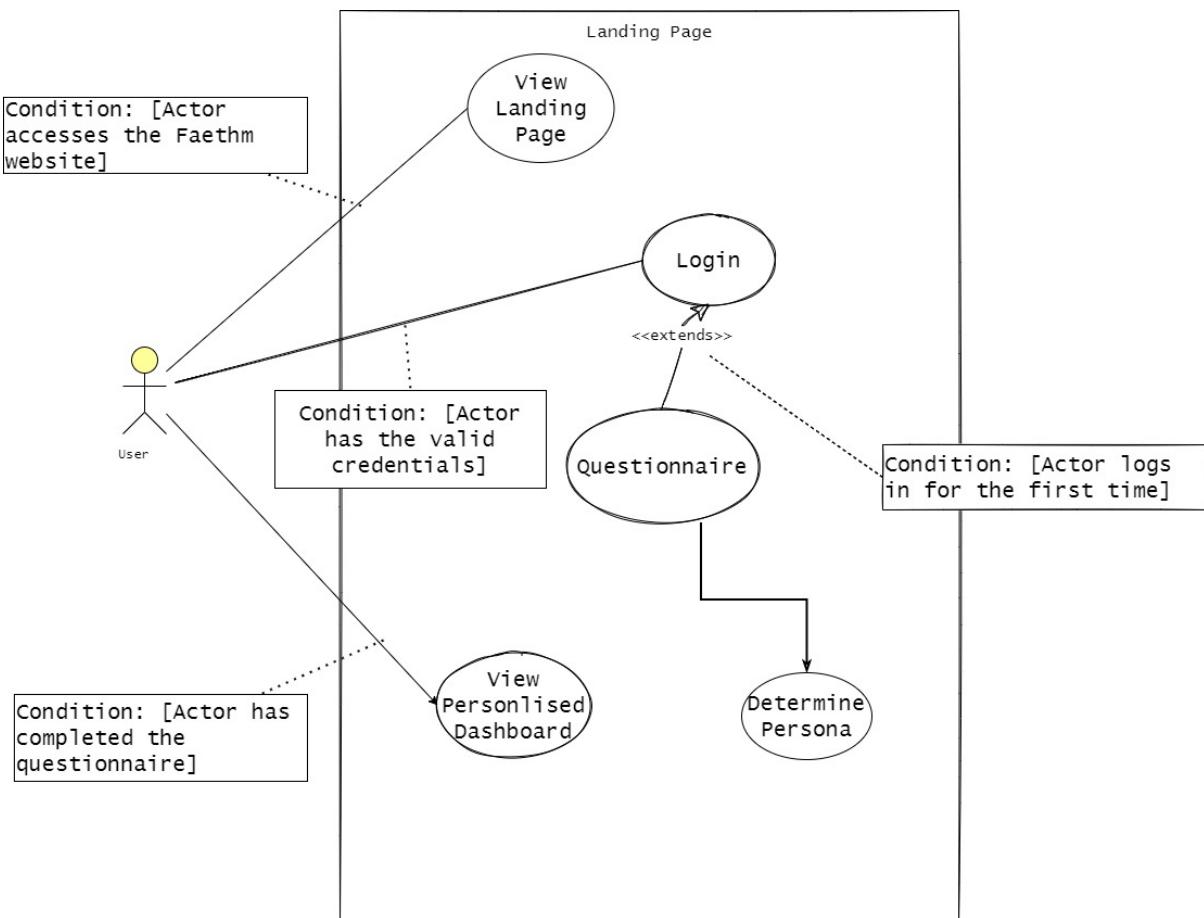
The feedback provided was overwhelmingly positive. From a design perspective, Karen has approved the Onboarding UI's overall design, user-friendliness, and intuitiveness. From a code perspective, Shu has approved of its readability, interpretability, and structure. Additionally, Shu expressed her appreciation of the team's ability to follow-through on her suggestions.

The Faethm team also congratulated our team for the high standard of Deliverables we have produced over the course of this PACE project.

### **Team Response**

The team is delighted and encouraged with the feedback provided. With the design and, now, code concerns addressed, the team will focus on producing tests.

### 3. Use Case Diagram



## 4. Use Case Descriptions and User Stories

Kindly note, while there are nine Personas, each use-case, success conditions and failed conditions remains the same for first-time users. Additionally, the Onboarding UI will only be served to first-time users (FTU).

### 4.1 Login

<b>Use Case</b>	Login
<b>User Story</b>	As a User, I want to login to the Faethm application.
<b>Preconditions</b>	The User has a valid email and password. The User has internet access and a web browser.
<b>Success Condition</b>	The User is able to login and access the Onboarding UI.
<b>Failed Condition</b>	The User is unable to login.
<b>Actors</b>	Primary Actor – User Secondary Actor – Faethm Application
<b>Trigger</b>	The User launches the Onboarding UI using their web browser
<b>Description</b>	<ol style="list-style-type: none"><li>1. The Onboarding UI serves a login form. If the application is not available or fails to load, then refer to <b>Alternative Flow 1 “System Under Maintenance”</b>.</li><li>2. The user inputs their email address and password.</li><li>3. The application authenticates the user's credentials.</li><li>4. If the application rejects the entered credentials, then refer to <b>Alternative Flow 2 “Invalid Credentials”</b>.</li></ol>
<b>Alternative Flows</b>	<ol style="list-style-type: none"><li>1. System Under Maintenance<ol style="list-style-type: none"><li>a. Display an error message “System under maintenance”.</li></ol></li><li>2. Invalid Credentials<ol style="list-style-type: none"><li>a. Display an error message “Invalid Email or Password”.</li></ol></li></ol>

## 4.2 Landing Page

<b>Use Case</b>	View Landing page
<b>User Story</b>	As a User, I want to be greeted by a minimal, aesthetic landing page that eases me into the onboarding process.
<b>Preconditions</b>	The User has a valid email and password. The User has internet access and a web browser.
<b>Success Condition</b>	The User can view the landing page.
<b>Failed Condition</b>	The User cannot view the landing page.
<b>Actors</b>	Primary Actor – User Secondary Actor – Faethm Application
<b>Trigger</b>	The User logs into the Onboarding UI for the first time.
<b>Description</b>	<ol style="list-style-type: none"><li>1. The Onboarding UI serves the landing splash page.</li><li>2. When the User clicks 'Next', proceed to <b>Use Case "Persona Page"</b>.</li></ol>
<b>Alternative Flows</b>	

#### 4.3 Persona Page

<b>Use Case</b>	Pick Persona
<b>User Story</b>	As a User, I want to check out the Faethm Application to see if it assists my business decisions in any way.
<b>Preconditions</b>	The User has a valid email and password. The User has internet access and a web browser. The User is a new user, that is, a first-time user.
<b>Success Condition</b>	The User is able to pick a persona that matches their position.
<b>Failed Condition</b>	The User does not select a Persona.
<b>Actors</b>	Primary Actor – User Secondary Actor – Faethm Application
<b>Trigger</b>	The User logs into the Onboarding UI and moves past the Landing Page.
<b>Description</b>	<ol style="list-style-type: none"><li>1. The Onboarding UI presents a form to determine the persona of the employee.</li><li>2. The User has to pick from the variety of personas presented. If the User does not pick an answer, refer to <b>Alternative Flow 1 “Answer(s) Required”</b></li></ol>
<b>Alternative Flows</b>	<ol style="list-style-type: none"><li>1. Answer(s) Required<ol style="list-style-type: none"><li>a. Display a warning modal</li></ol></li></ol>

#### 4.4 Survey Page

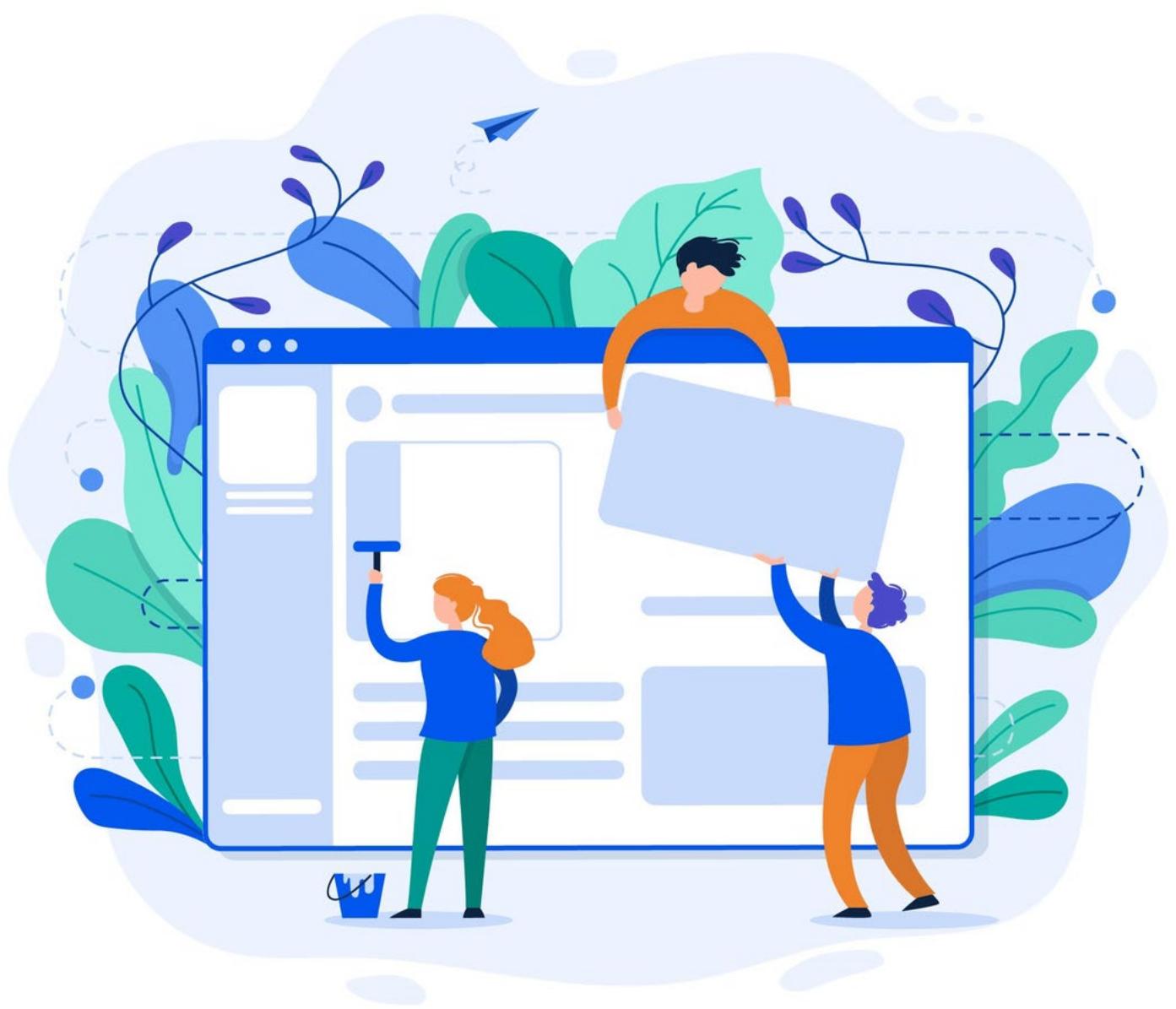
<b>Use Case</b>	Gather user information on use-case
<b>User Story</b>	As a User, I want to check out the Faethm Application to see if it assists my business decisions in any way.
<b>Preconditions</b>	The User has a valid email and password. The User has internet access and a web browser. The User is a new user, that is, a first-time user.
<b>Success Condition</b>	The User answers all questions on the short survey.
<b>Failed Condition</b>	The User does not answer all questions.
<b>Actors</b>	Primary Actor – User Secondary Actor – Faethm Application
<b>Trigger</b>	The User chooses a Persona and clicks ‘Next’.
<b>Description</b>	<ol style="list-style-type: none"><li>1. The Onboarding UI serves a survey that the User fills out.</li><li>2. The User answers all questions.</li></ol> <p>If the User leaves any question unanswered, refer to <b>Alternative Flow 1 “Answers Required”</b></p>
<b>Alternative Flows</b>	<ol style="list-style-type: none"><li>2. Answers Required<ol style="list-style-type: none"><li>a. Display a warning modal</li></ol></li></ol>

## 4.5 Personalised Dashboard

<b>Use Case</b>	View Personalised Dashboard
<b>User Story</b>	As a User, I want to review my company's numbers and make decisions in correspondence to achieve my goals.
<b>Preconditions</b>	The User has a valid email and password The User has internet access and a web browser.
<b>Success Condition</b>	The User views their personalized dashboard and access features provided by the application.
<b>Failed Condition</b>	
<b>Actors</b>	Primary Actor – User Secondary Actor – Faethm Application
<b>Trigger</b>	The User answers and submits the survey.
<b>Description</b>	<ol style="list-style-type: none"> <li>1. The User logs into the Faethm application. Refer to use case <b>Login</b> to view how a user logs in.</li> <li>2. The User picks a persona if they are logging in for the first time</li> <li>3. Review use case description 'Pick Persona' to see how persona is picked</li> <li>4. The User will be presented with a loading page while the data from the questionnaire is processed.</li> <li>5. After processing is complete, the User will be presented with select features of the Faethm application along with relevant articles, and statistics determined through the questionnaire.</li> </ol>
<b>Alternative Flows</b>	

## Assumptions

- A1. We will be able to maintain contact with the representatives from Faethm for the duration of the project.
- A2. Faethm will provide us with the appropriate resources to complete the project.
- A3. All group members have access to appropriate hardware and software to complete the project.
- A4. The project's requirements can be changed by Faethm at any point in the project if it is required.
- A5. The project's requirements can be changed by Faethm at any point in the project if it is required.
- A6. Access to the existing CSS styles to ensure a consistent UI with the rest of the app.
- A7. While Faethm has the flexibility to accept a number of tech stacks, they ultimately prefer MERN (MongoDB, Express, React, Node). This report assumes that MERN will be used.
- A8. The solution will be provided as a stand-alone single-page application (SPA) separate from the existing application.
- A9. A laptop use case for most users has been assumed with mobile-friendliness, while ideal if achieved, as an added benefit.
- A10. Access will not be granted to Faethm's existing code base for security.
- A11. Faethm will provide us with the assistance necessary to develop this solution.



# Design Document

Lance Te, Sepehr Torfeh Nejad, Marcus Ikeda, Erik Horvath, Rojwal Shrestha



# Contents

<b>1. System Design Document .....</b>	<b>2</b>
1.1 System Architecture .....	2
1.2 Package Diagram .....	2
1.3 Storage/ Persistent Data Strategy .....	3
1.4 Concurrent Processes .....	3
1.5 User Interface Strategy.....	3
1.6 Design Choices and Trade-offs .....	3
<b>2. User Interface Layouts .....</b>	<b>4</b>
2.1 Login Page.....	4
2.2 Landing Page.....	4
2.3 Persona Page .....	5
2.4 Survey Page .....	5
2.5 Loading Page.....	6
<b>3. Window Navigation Diagram .....</b>	<b>7</b>
<b>4. Data Definitions / ER Diagram.....</b>	<b>8</b>
4.1 Overview.....	8
4.2 ER Diagram .....	8
4.3 Data Types .....	9
4.3.1 User Collection.....	9
4.3.2 Persona Collection.....	9
4.3.3 Use-Case Collection .....	9
<b>5. Activity Diagrams .....</b>	<b>10</b>
5.1 Login .....	10
5.2 Survey .....	10
5.3 Information Handling .....	11

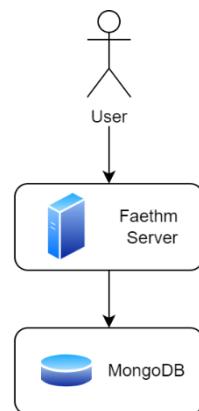
# 1. System Design Document

## 1.1 System Architecture

The Onboarding UI has a simple, traditional system architecture. As it will sit in front of the current Faethm web application, it will be hosted on the Faethm servers.

When users access the Faethm app, they are served the login page where their identity is authenticated before being served the Onboarding UI. Each user's credentials are stored in the Faethm MongoDB.

As users interact with the Onboarding UI, the information gathered will be sent back to the Faethm server and stored as part of their document record in the MongoDB.

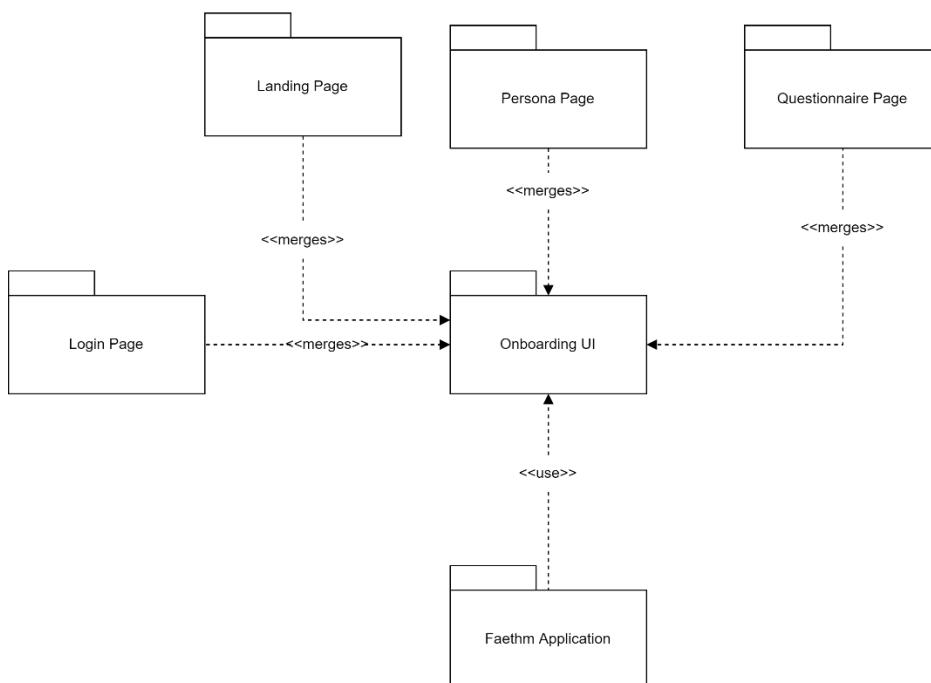


## 1.2 Package Diagram

The package diagram shows the organisation and arrangement of the various models / elements / components of a software application in the form of packages. The Onboarding UI is the central model element that merges the:

- Login page - Processes user authentication
- Landing Page – Aesthetically pleasing landing page to ease users into the onboarding process
- Persona page - Determines the user's persona
- Survey page - Determines the user's use case

Additionally, the remaining Faethm application uses the Onboarding UI as the first-time user experience (FTUE) before directing users to the primary application.



### **1.3 Storage/ Persistent Data Strategy**

Our Onboarding UI collects and stores a range of persistent data that tracks user information. The initial user login credentials are stored upon creating an account and following this, the user is prompted to complete our onboarding UI. This process collects additional information about the user which is converted to data that is stored in the database. Our team has developed a process to manage the storage of this data and to strategically balance the durability of the data with the appropriate performance level when retrieving this data to ensure a seamless user experience.

We have decided to utilise the MongoDB cloud-based database application to both manage the database and to store the user data. MongoDB is an industry trusted SaaS application specialising in highly customisable document-oriented NoSQL database storage. MongoDB has pre-existing best practices and automates certain operations to ensure that data is kept secure, and the database runs smoothly. Cloud based storage means that it is fast to access anywhere as it reduces our hardware costs and running costs of on-premise servers and machines.

### **1.4 Concurrent Processes**

The onboarding UI tool is a linear process that doesn't require any simultaneous processes. Multiple concurrent user connections are managed due to NodeJS being single threaded which means that it takes all simultaneous requests, but it will process each individually and one after the other. Multiple users accessing the database at the same time is managed automatically by MongoDB as it has been created to do so and is operating in a cloud environment.

### **1.5 User Interface Strategy**

As our onboarding UI requires the user to interact with the application in order to gather information, we have developed a plan of action to improve the user experience and ensure efficient collection of data. Our application follows the JWS open standard to securely transmit information between the users and the application as well as utilising the Redux JavaScript library to manage the application state. The interactive GUI is created with a responsive CSS design that allows users to share their information by clicking on buttons once prompted.

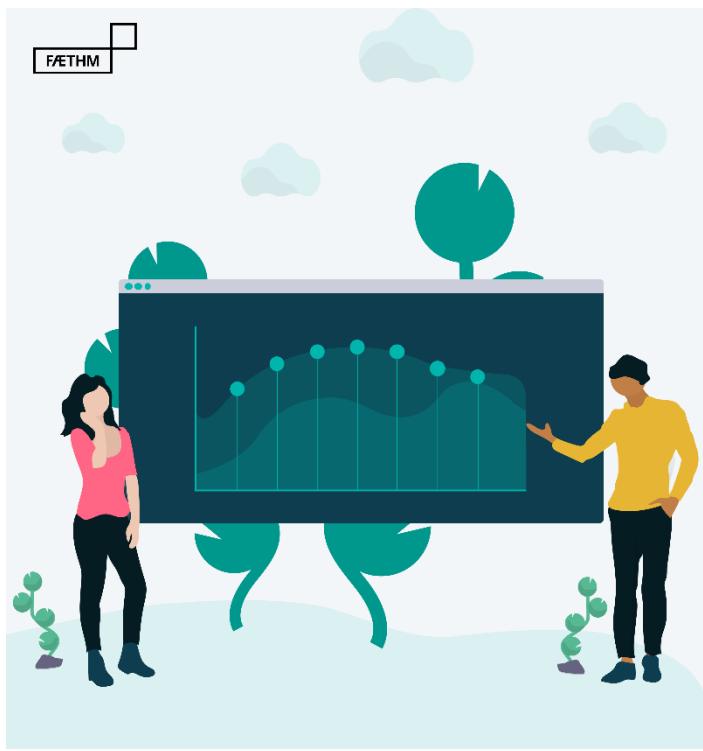
### **1.6 Design Choices and Trade-offs**

Our application was designed using the MERN stack rather than a more traditional PHP + jQuery for easier implementation with existing Faethm AI application. Our colour schemes and icons are created to match the existing Faethm AI colour scheme and existing design format. The specific user categories were provided to us by the Faethm team and we built our application around this template. We have also added on an additional survey to further categorise the user which is designed to create a greater personalised experience to the user. The survey follows the previous pages design scheme and the landing page design scheme from the Faethm site.

## 2. User Interface Layouts

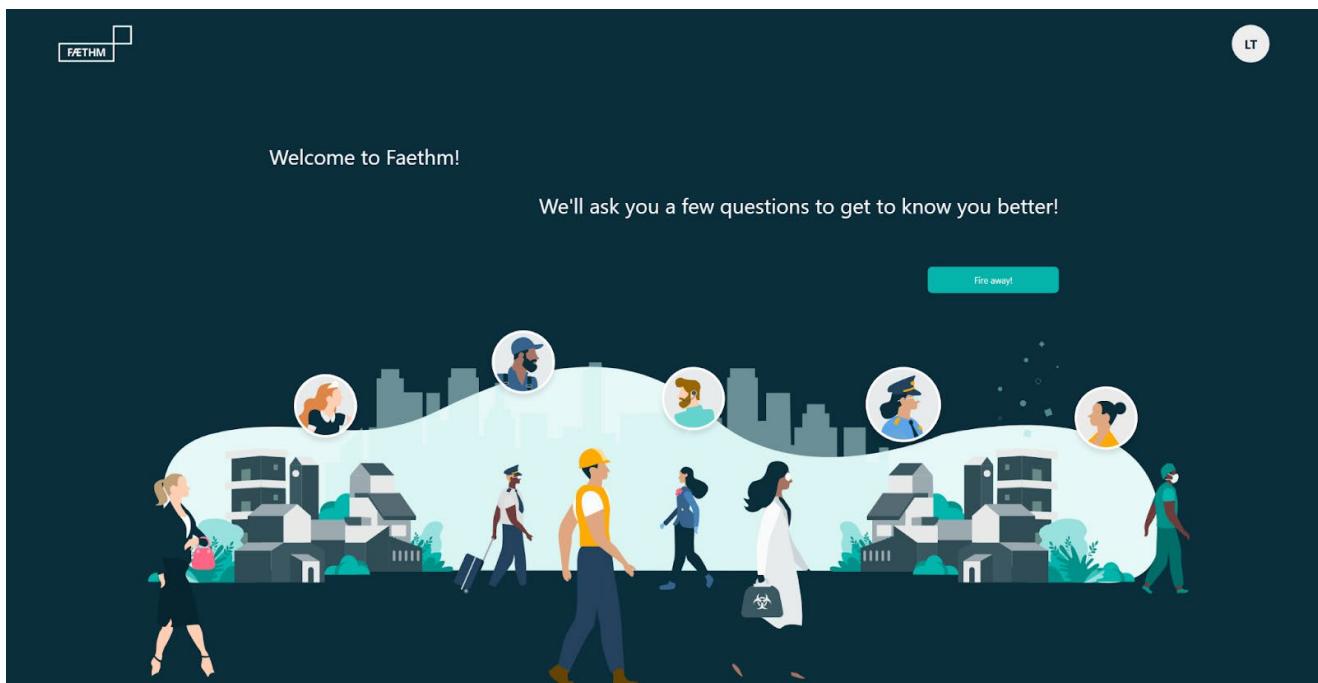
The following images is the linear progression of the user interface layouts as the user progresses through the onboarding process.

### 2.1 Login Page



The illustration shows a woman in a pink top and black pants standing next to a large digital screen. A man in a yellow shirt and dark pants is pointing at the screen. The screen displays a teal-colored line graph with several data points. The background features stylized green trees and clouds against a light blue sky. In the top left corner, there is a small logo with the word "FAETHM". To the right of the illustration, the word "Welcome" is displayed in a bold, sans-serif font. Below it, the text "Log in to your Faethm account" is written in a smaller font. The main form area contains two input fields: "Email address" and "Password", each with a placeholder text "Type in email address" and "Type in password" respectively. Below the password field is a "Forgot password?" link. At the bottom of the form is a teal-colored "Log in" button.

### 2.2 Landing Page



The illustration depicts a diverse group of people walking across a bridge over a river, set against a dark blue background with a city skyline silhouette. Above the bridge, the text "Welcome to Faethm!" is displayed in white. Below it, the text "We'll ask you a few questions to get to know you better!" is shown. In the top right corner, there is a circular icon with the letters "LT". At the bottom right of the main content area, there is a teal-colored "Find out!" button. The overall theme is one of community and diversity.

## 2.3 Persona Page

Help us customise your experience!  
Please select which category best describes you.



Executive



Manager



Workforce Planner



Head of Division



Employee



HR Manager



Data Analyst



Learning Manager



Automation Engineer

[Next](#)

## 2.4 Survey Page

1. To what extent do you plan to maintain and upskill your current workforce?

Low

Medium

High

2. To what extent is your workforce willing to learn new skills and undertake further studies?

Low

Medium

High

3. To what extent do you believe your industry will be impacted by emerging technologies?

Low

Medium

High

4. To what extent do you think your industry will be impacted by automation?

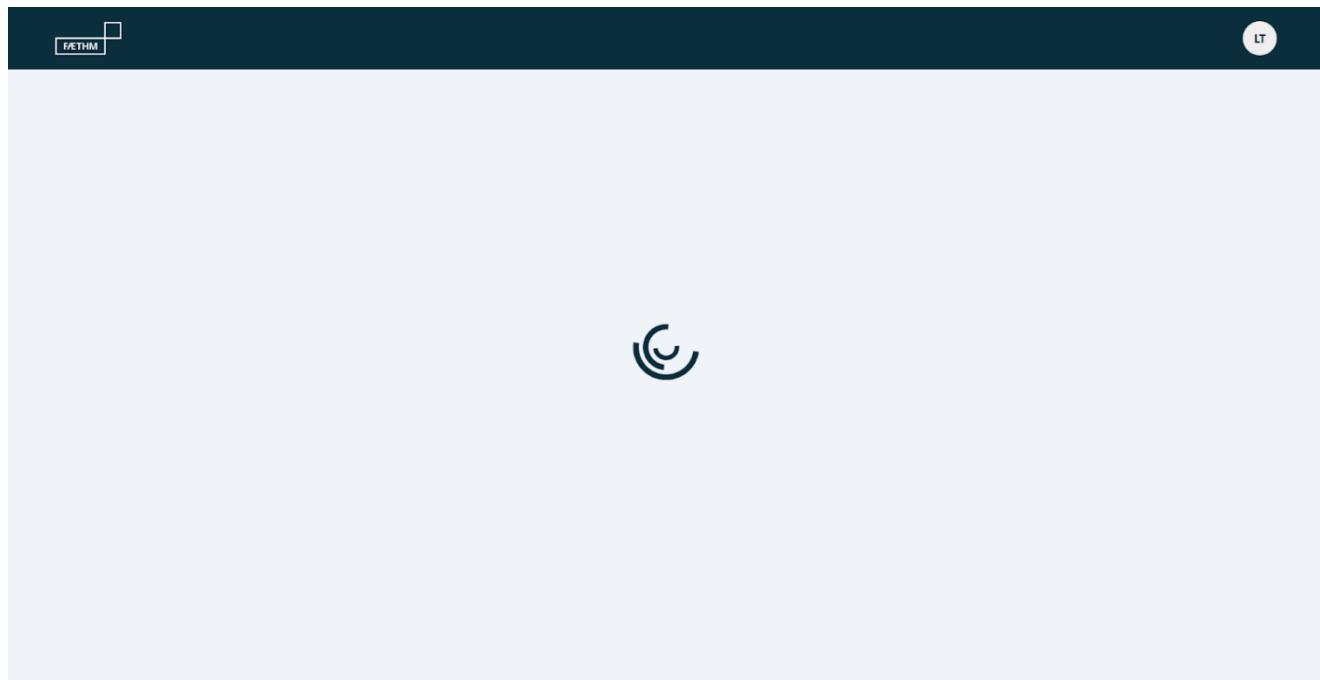
Low

Medium

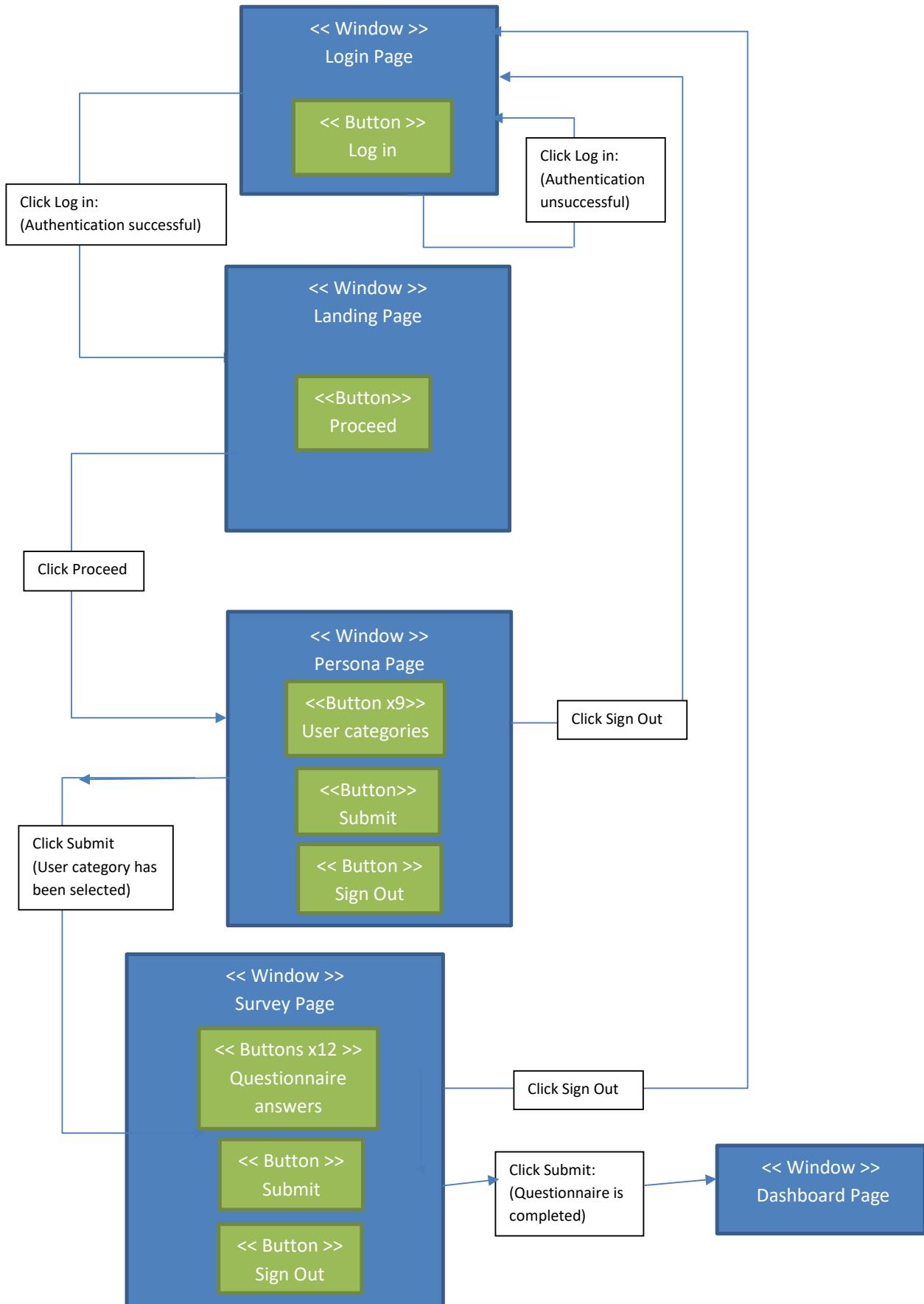
High

[Submit](#)

## 2.5 Loading Page



### 3. Window Navigation Diagram



## 4. Data Definitions / ER Diagram

### 4.1 Overview

The database is structured into three collections: Users, Personas, Use-Case.

The User collection contains an individual user's details including their first name, last name, email, password and also store the Persona they select and Use-Case allocation following their onboarding through the Onboarding UI.

The Persona collection contains the various pre-defined categories a user will fall under, these are:

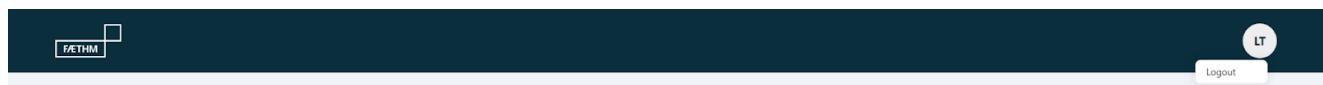
- Executive
- Head of Division
- Data Analyst
- Manager
- Employee
- Learning Manager
- Workforce Planner
- HR Manager
- Automation Engineer

The Use-Case collection contains the various pre-defined categories a user will fall under, these are:

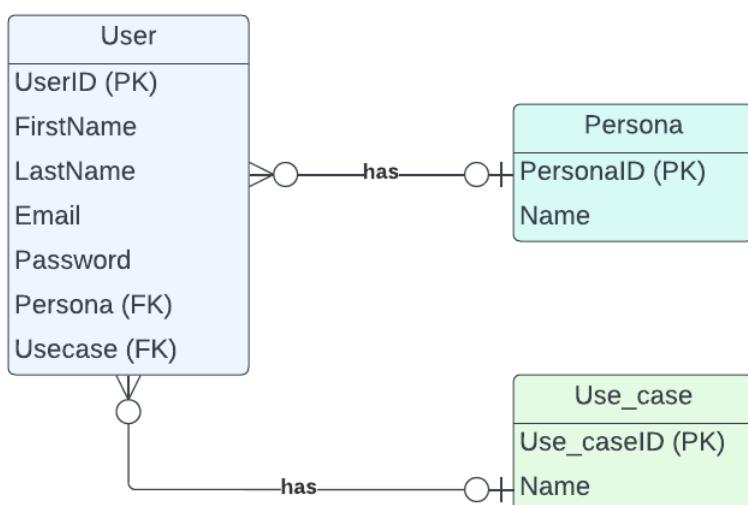
- Workforce Insight
- Technology Insight
- Job Insight
- Job Profiles
- Job Corridor

Both the Persona ID and Use-Case ID selected by the user and determined from the survey respectively, will be stored in their corresponding User document. Beyond the onboarding, first-time user experience, this data will be used to enable personalisation of their experience using the Faethm application.

In terms of a directly applicable use of the data, the user's first name and last name are used as the default user icon. This is pictured in the top right corner of the navigation bar below.



### 4.2 ER Diagram



## 4.3 Data Types

### 4.3.1 User Collection

Name	Description	Data Type	Required
id	Unique identifier for every user	ObjectID	Y
firstname	User's first name	String	Y
lastname	User's last name	String	Y
email	User's email address	String	Y
password	User's password hashed using bcrypt(10)	String	Y
persona	Persona ID linked to the corresponding document in the Persona collection	String	N
use_case	Use-Case ID linked to the corresponding document in the Use-Case collection	String	N
created_at	Timestamp of when the user was created	Timestamp	Y
updated_at	Timestamp of when the user was last updated	Timestamp	N

### 4.3.2 Persona Collection

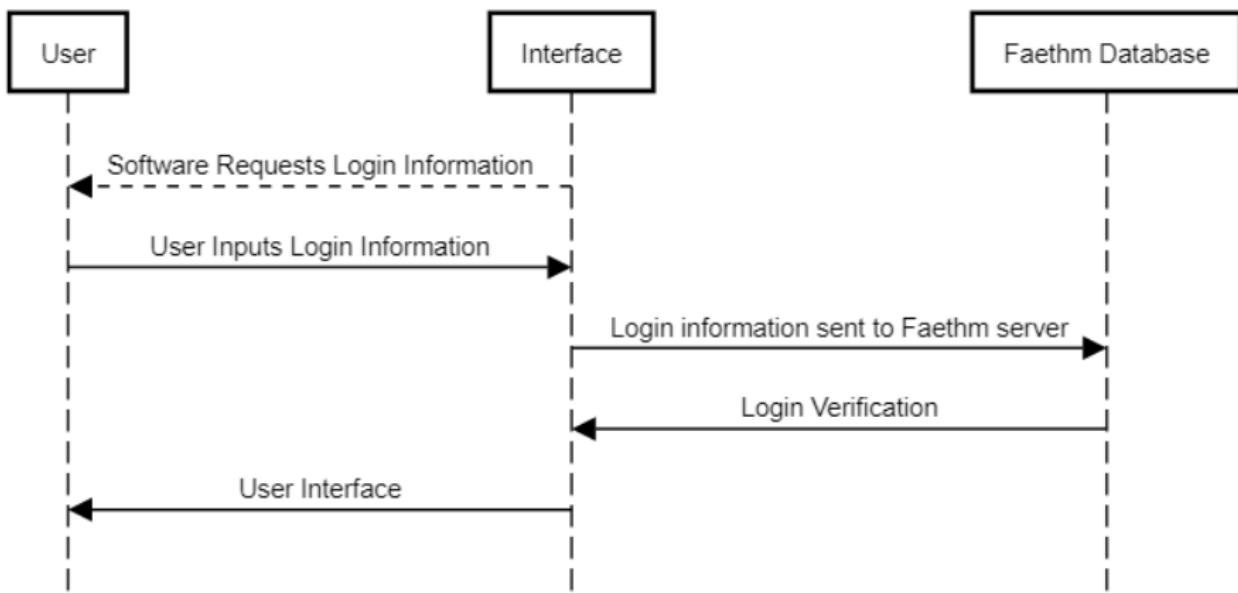
Name	Description	Data Type	Required
id	Unique identifier for every Persona	ObjectID	Y
name	The name label of each Persona	String	Y

### 4.3.3 Use-Case Collection

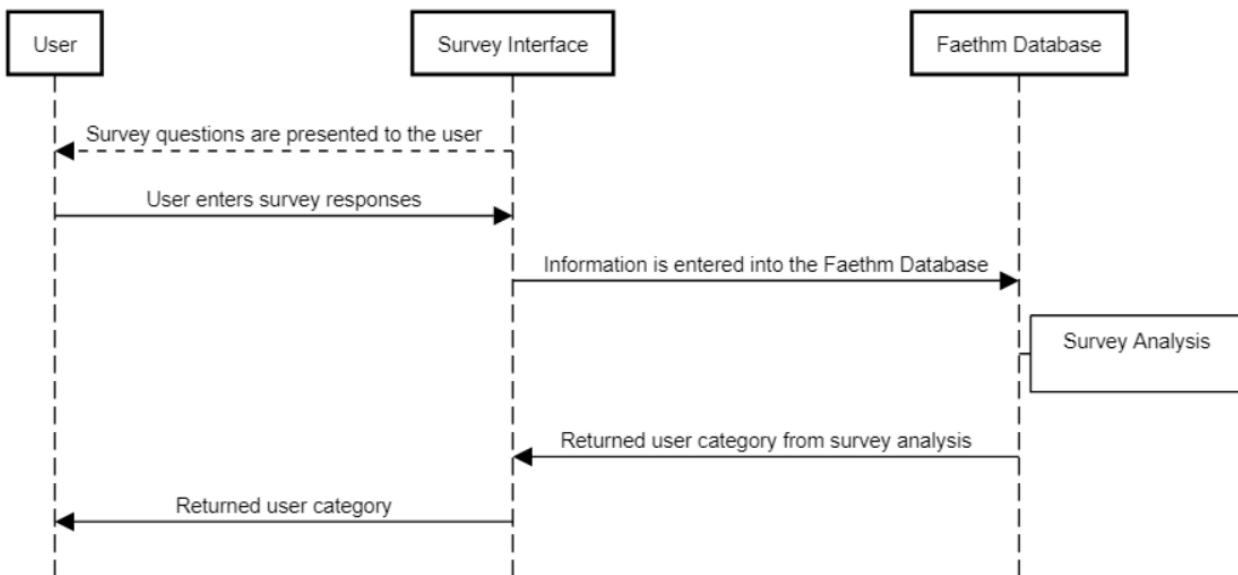
Name	Description	Data Type	Required
id	Unique identifier for every Use-Case	ObjectID	Y
name	The name label of each Use-Case	String	Y

## 5. Activity Diagrams

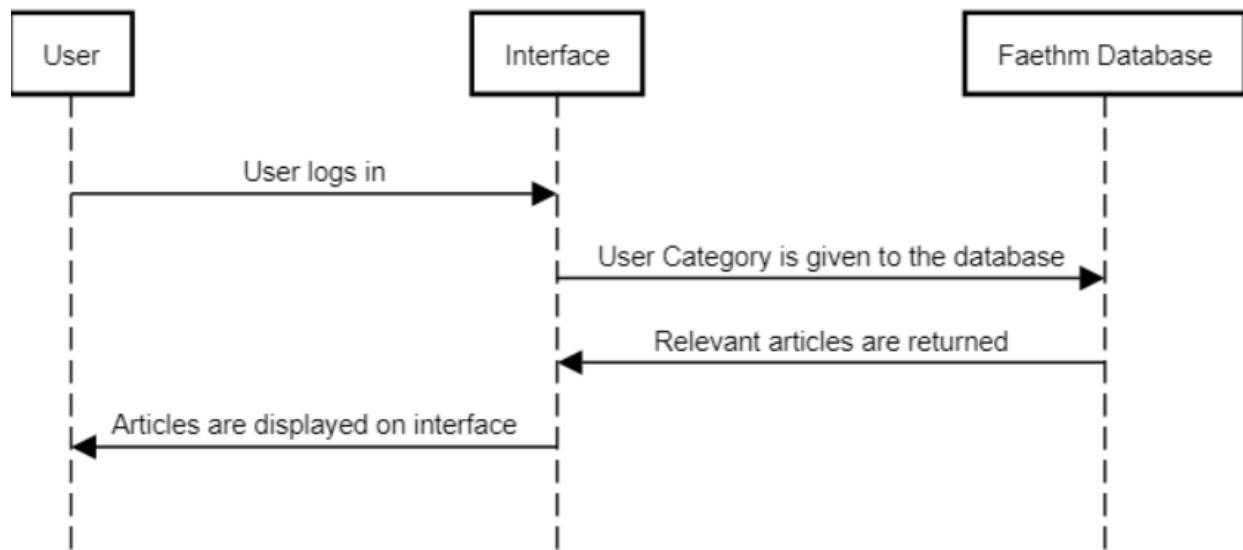
### 5.1 Login



### 5.2 Survey



### 5.3 Information Handling





# Testing Document

Lance Te, Sepehr Torfeh Nejad, Marcus Ikeda, Erik Horvath, Rojwal Shrestha

FÆTHM

## Contents

<b>1. Test Plan .....</b>	<b>2</b>
1.1 Testing Strategy .....	2
1.2 Test Types .....	2
1.3 Test Schedule .....	3
1.4 Test Tools and Resources .....	3
1.5 Test Deliverables .....	3
1.6 Test Resourcing .....	3
1.7 Test Milestones .....	3
<b>2. Test Case Specifications .....</b>	<b>4</b>
2.1 Login Functionality .....	4
2.2 Landing Page Functionality .....	5
2.3 Survey Functionality .....	6
2.4 Security Functionality .....	8
2.5 User Interface Functionality .....	9
2.6 User Experience .....	15

# 1. Test Plan

## 1.1 Testing Strategy

As our Onboarding UI is not hosted on the cloud and we have chosen to instead host it on GitHub, this means that testing will need to be carried out on each developer's local machine. Whenever a developer wants to test code, the updated version of the code base will need to be pulled and fetched from the remote repository. As bugs and errors are fixed, and new code is added, the developer will then push their local changes to the remote repository for other developers to perform further testing and verification. Every feature will be implemented on a separate branch in the remote repository to ensure the main code base remains stable at all times. Further, it mitigates the risk of errors and allows for easier debugging since any known errors following a merge with the main branch can be easily identified as coming from the feature branch.

When testing the Onboarding UI for the first time, developers need to clone the environment by:

1. Cloning the remote repository.
2. Using node package manager (npm) to install all dependencies on their local machine.
3. Setting up the .env file to define environment variables including port, MongoDB URL, and JWT secret.
4. Running the command `npm run dev`.

The developer will then be able interact with the Onboarding UI and perform the tests outlined in the [Test Specification](#) section below.

With the heavy user-interaction expected on the Onboarding UI, compared to its relatively low functionality, most tests will fall under black-box tests that are designed with the end-user in mind. White-box testing will also be performed on the back-end system to evaluate and examine the code; ultimately to verify the internal structures are working as intended.

## 1.2 Test Types

The test types will predominantly fall under system tests as the Onboarding UI will be providing the first-time user experience (FTUE) for the current Faethm web application.

However, as new features and functionalities are designed and implemented, the team will undergo rigorous regression testing to ensure that it works as intended and the existing features and functionalities are not broken. The Onboarding UI includes a login feature which requires penetration testing to ensure that users are authenticated and can only use features they are authorised to access.

We will also perform in-depth testing of the UI and UX system where we will be testing the platform on multiple browsers and screen sizes to ensure it is compatible with all devices. We will also be gathering feedback on user experience from people who are not part of the design process or stakeholders in the application to get an outsider perspective for the overall feel of the application.

The data collected from these tests will be recorded by our team in the process of running the tests. We will be looking ultimately to see if the operation was successful or not. From the test users who are testing the UX component, we will provide a quick rating survey to rate the experience and the usability from 1-5 as well as leaving space to provide their own written feedback or suggestions.

### **1.3 Test Schedule**

Given the agile methodology used throughout this project, work is delivered in small increments. Thus, the team does not have a specific test schedule. Instead, as the team implements new features and functionalities, testing will be done immediately after to ensure that it works as intended. This iterative, incremental approach means that our suite of test specifications will grow. These changes will be recorded and included for an updated Testing Document (Increment 2).

In the week before the presentation and handover of the Onboarding UI to Faethm on Thursday 2nd June 2022, the team will undergo a thorough test of the entire web application in accordance with the Testing Specifications. During this week we will also have the user experience tested with external users.

### **1.4 Test Tools and Resources**

There are a range of tools available for testing beyond those mentioned below. However, these are agreed minimum standards that all developers will adhere to. Any additional tools, browser extensions, IDE extensions, and other resources used is up to each developer's discretion.

- **Typescript:** While Typescript is not a replacement for tests, it is a good tool to catch errors and ensure static checking before reaching production.
- **React Dev Tools:** Provides the developer with the React component tree and an interface to inspect and interact with elements including components, states, and stores.
- **Jest:** Provides the developer with a JavaScript library designed specifically for React to create, run, and structure tests.
- **Seeder script:** Populates database tables/collections with test data.
- **npm run dev:** Runs the development environment locally for the developer to perform tests on the Onboarding UI.

### **1.5 Test Deliverables**

The main test deliverable to the client is this Testing Document. However, as new features are implemented, the team will provide updated versions of this Testing Document.

### **1.6 Test Resourcing**

The tests included in the [Test Specification](#) below outline which developers are responsible for which test(s).

### **1.7 Test Milestones**

The discussion and addition of new features that results from the agile methodology will serve as a testing milestone. This involves implementing the feature and subsequently testing to ensure the feature works appropriately and interacts with the rest of the existing application cohesively.

The week before the presentation and handover of the Onboarding UI to Faethm on Thursday 2nd June 2022 remains the main test milestone the Onboarding UI team continues to work towards for all technical code, UI and UX testing.

## 2. Test Case Specifications

### 2.1 Login Functionality

<b>Test Case ID</b>	LOGIN_001
<b>Test Designed by</b>	Sep
<b>Test Reviewed by</b>	Lance
<b>Test Priority</b>	Medium
<b>Test Scenario</b>	Test login page by verifying successful user login with valid username and password.
<b>Preconditions</b>	<ol style="list-style-type: none"><li>1. Tester has pulled the latest version from GitHub</li><li>2. Database is seeded using the seeder script</li><li>3. Application is running locally using <code>npm run dev</code> command</li></ol>
<b>Test Steps</b>	<ol style="list-style-type: none"><li>1. Enter email address</li><li>2. Enter password</li><li>4. Click the 'Login' button</li></ol>
<b>Successful Outcome</b>	An authorised user can login successfully with their credentials. The JWT is created and returned upon successful user login and stored in local storage. Invalid user credentials will return an error and user can't login.

<b>Test Executed by</b>	<Name>
<b>Test Execution date</b>	<Date>
<b>Test Status</b>	Not Executed
<b>Test Log</b>	<Log>

## 2.2 Landing Page Functionality

<b>Test Case ID</b>	LANDING_PAGE_001
<b>Test Designed by</b>	Sep
<b>Test Reviewed by</b>	Lance
<b>Test Priority</b>	Low
<b>Test Scenario</b>	The user is served a landing page that welcomes them into the onboarding process.
<b>Preconditions</b>	<ol style="list-style-type: none"><li>1. Tester has pulled the latest version from GitHub</li><li>2. Database is seeded using the seeder script</li><li>3. Application is running locally using <code>npm run dev</code> command</li><li>4. User is logging in for the first time</li></ol>
<b>Test Steps</b>	<ol style="list-style-type: none"><li>1. Log into the Onboarding UI (Login Functionality)</li><li>2. Wait for the landing page CSS animations to render</li><li>3. Click the 'Fire Away' button</li></ol>
<b>Successful Outcome</b>	The user should be routed to the Persona page.

<b>Test Executed by</b>	<Name>
<b>Test Execution date</b>	<Date>
<b>Test Status</b>	Not Executed
<b>Test Log</b>	<Log>

## 2.3 Survey Functionality

<b>Test Case ID</b>	SURVEY_001
<b>Test Designed by</b>	Sep
<b>Test Reviewed by</b>	Lance
<b>Test Priority</b>	High
<b>Test Scenario</b>	The user can choose a persona from the 9 predefined personas.
<b>Preconditions</b>	<ol style="list-style-type: none"><li>1. Tester has pulled the latest version from GitHub</li><li>2. Database is seeded using the seeder script</li><li>3. Application is running locally using <code>npm run dev</code> command</li><li>4. User is logging in for the first time</li></ol>
<b>Test Steps</b>	<ol style="list-style-type: none"><li>1. Select a persona</li><li>2. Click the 'Next' button</li></ol>
<b>Successful Outcome</b>	The ID of the selected persona is stored in the user's document in the User collection and the user is routed to the survey page.

<b>Test Executed by</b>	<Name>
<b>Test Execution date</b>	<Date>
<b>Test Status</b>	Not Executed
<b>Test Log</b>	<Log>

<b>Test Case ID</b>	SURVEY_002
<b>Test Designed by</b>	Sep
<b>Test Reviewed by</b>	Lance
<b>Test Priority</b>	High
<b>Test Scenario</b>	The user answers a small number of multiple-choice questions.
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Tester has pulled the latest version from GitHub</li> <li>2. Database is seeded using the seeder script</li> <li>3. Application is running locally using <code>npm run dev</code> command</li> <li>4. User is logging in for the first time</li> </ol>
<b>Test Steps</b>	<ol style="list-style-type: none"> <li>1. For each question, select Low, Medium, or High</li> <li>2. Click the 'Submit' button</li> </ol>
<b>Successful Outcome</b>	The values of Low (0), Medium (1), High (2) are stored in the user's document in the User collection and the user is served a short loading screen.

<b>Test Executed by</b>	<Name>
<b>Test Execution date</b>	<Date>
<b>Test Status</b>	Not Executed
<b>Test Log</b>	<Log>

## 2.4 Security Functionality

Test Case ID	SECURITY_001
Test Designed by	Sep
Test Reviewed by	Lance
Test Priority	High
Test Scenario	Testing protected routes on the application. Users can not access survey page without first being authenticated for security reasons.
Preconditions	
Test Steps	Attempt accessing any of the views via direct URL, for example, <a href="http://localhost:3000/landing-page">http://localhost:3000/landing-page</a>
Successful Outcome	The user is redirected back to the login page.

Test Executed by	<Name>
Test Execution date	<Date>
Test Status	Not Executed
Test Log	<Log>

## 2.5 User Interface Functionality

<b>Test Case ID</b>	UI_001
<b>Test Designed by</b>	Erik
<b>Test Reviewed by</b>	Lance
<b>Test Priority</b>	Medium
<b>Test Scenario</b>	Test the application compatibility with major browsers and zoom levels.
<b>Preconditions</b>	<ol style="list-style-type: none"><li>1. Tester has pulled the latest version from GitHub</li><li>2. Database is seeded using the seeder script</li><li>3. Application is running locally using <code>npm run dev</code> command</li><li>4. Tester has all major browsers installed including: Google Chrome, Mozilla Firefox, Microsoft Edge, Safari</li></ol>
<b>Test Steps</b>	<ol style="list-style-type: none"><li>1. Using any of the listed browsers and 100% zoom scale.</li><li>2. Perform all <b>Survey Functionality</b> tests</li><li>3. Repeat steps 1-3 using 50% zoom and 150% zoom.</li><li>4. Repeat steps 1-4 until all browsers are tested.</li></ol>
<b>Successful Outcome</b>	

<b>Test Executed by</b>	<Name>
<b>Test Execution date</b>	<Date>
<b>Test Status</b>	Not Executed
<b>Test Log</b>	<Log>

<b>Test Case ID</b>	UI_002
<b>Test Designed by</b>	Lance
<b>Test Reviewed by</b>	Sep
<b>Test Priority</b>	Low
<b>Test Scenario</b>	User interacting with form text inputs.
<b>Preconditions</b>	
<b>Test Steps</b>	The user hovers over form text inputs.
<b>Successful Outcome</b>	The input border should darken with an ease-in transition.

<b>Test Executed by</b>	<Name>
<b>Test Execution date</b>	<Date>
<b>Test Status</b>	Not Executed
<b>Test Log</b>	<Log>

<b>Test Case ID</b>	UI_003
<b>Test Designed by</b>	Lance
<b>Test Reviewed by</b>	Sep
<b>Test Priority</b>	Low
<b>Test Scenario</b>	User interacting with buttons
<b>Preconditions</b>	
<b>Test Steps</b>	The user hovers over button.
<b>Successful Outcome</b>	The button background colour should lighten with an ease-in transition.

<b>Test Executed by</b>	<Name>
<b>Test Execution date</b>	<Date>
<b>Test Status</b>	Not Executed
<b>Test Log</b>	<Log>

<b>Test Case ID</b>	UI_004
<b>Test Designed by</b>	Lance
<b>Test Reviewed by</b>	Sep
<b>Test Priority</b>	Low
<b>Test Scenario</b>	Card-flip animation.
<b>Preconditions</b>	
<b>Test Steps</b>	User hovers over Persona card.
<b>Successful Outcome</b>	The button background colour should lighten with an ease-in transition.

<b>Test Executed by</b>	<Name>
<b>Test Execution date</b>	<Date>
<b>Test Status</b>	Not Executed
<b>Test Log</b>	<Log>

<b>Test Case ID</b>	UI_005
<b>Test Designed by</b>	Lance
<b>Test Reviewed by</b>	Sep
<b>Test Priority</b>	Low
<b>Test Scenario</b>	User selection on the Persona Page and Survey Page.
<b>Preconditions</b>	
<b>Test Steps</b>	<ol style="list-style-type: none"> <li>1. The user selects the Persona that corresponds to their organisational position on the Persona Page</li> <li>2. The user selects one answer in each of the multiple-choice questions on the Survey Page.</li> </ol>
<b>Successful Outcome</b>	The background and border of the cards selected should darken compared to those unselected to indicate user selection.

<b>Test Executed by</b>	<Name>
<b>Test Execution date</b>	<Date>
<b>Test Status</b>	Not Executed
<b>Test Log</b>	<Log>

<b>Test Case ID</b>	UI_006
<b>Test Designed by</b>	Lance
<b>Test Reviewed by</b>	Sep
<b>Test Priority</b>	Low
<b>Test Scenario</b>	User menu.
<b>Preconditions</b>	
<b>Test Steps</b>	User hovers over their user icon in the navigation bar (top right corner looking at the entire page).
<b>Successful Outcome</b>	The user menu should appear below the user icon on hover with an ease-in transition.

<b>Test Executed by</b>	<Name>
<b>Test Execution date</b>	<Date>
<b>Test Status</b>	Not Executed
<b>Test Log</b>	<Log>

## 2.6 User Experience

<b>Test Case ID</b>	UX_001
<b>Test Designed by</b>	Erik
<b>Test Reviewed by</b>	Lance
<b>Test Priority</b>	Medium
<b>Test Scenario</b>	Provide an untrained user with the application and record user ratings on a scale of 1-5 regarding the user experience.
<b>Preconditions</b>	The Tester is external to the project but has access to a machine running the project. This limits the possibility to friends or relatives of the Onboarding UI team.
<b>Test Steps</b>	<ol style="list-style-type: none"><li>1. The user is provided access and login details to the application.</li><li>2. The user interacts with the application without instruction or guidance.</li><li>3. The user provides feedback regarding the overall experience, design, and ease-of-use.</li></ol>
<b>Successful Outcome</b>	The Tester provides positive feedback.

<b>Test Executed by</b>	<Name>
<b>Test Execution date</b>	<Date>
<b>Test Status</b>	Not Executed
<b>Test Log</b>	<Log>



# User Manual

Lance Te, Sepehr Torfeh Nejad, Marcus Ikeda, Erik Horvath, Rojwal Shrestha

FÆTHM

## Contents

<b>1. Introduction .....</b>	<b>2</b>
<b>2. Installation Guide .....</b>	<b>2</b>
2.1    Clone GitHub Repository .....	2
2.2    Installing Dependencies.....	3
2.2.1    Backend .....	3
2.2.2    Frontend .....	3
2.3    Seed Database .....	3
2.4    Run Application .....	3
<b>3. Configuration Settings .....</b>	<b>3</b>
<b>4. Usage .....</b>	<b>4</b>
4.1    End-User Usage .....	4
4.1.1    Login Page.....	4
4.1.2    Persona Page .....	5
4.1.3    Survey Page .....	5
4.2    Admin / Developer Usage.....	6
<b>5. Further Assistance .....</b>	<b>6</b>

# 1. Introduction

This user manual is aimed to provide documentation on the usage of the Onboarding UI for both end-users and admin/developer users. The purpose of this documentation involves the usage of the Onboarding UI involving its application and troubleshooting. It is aimed at showing the features of the application.

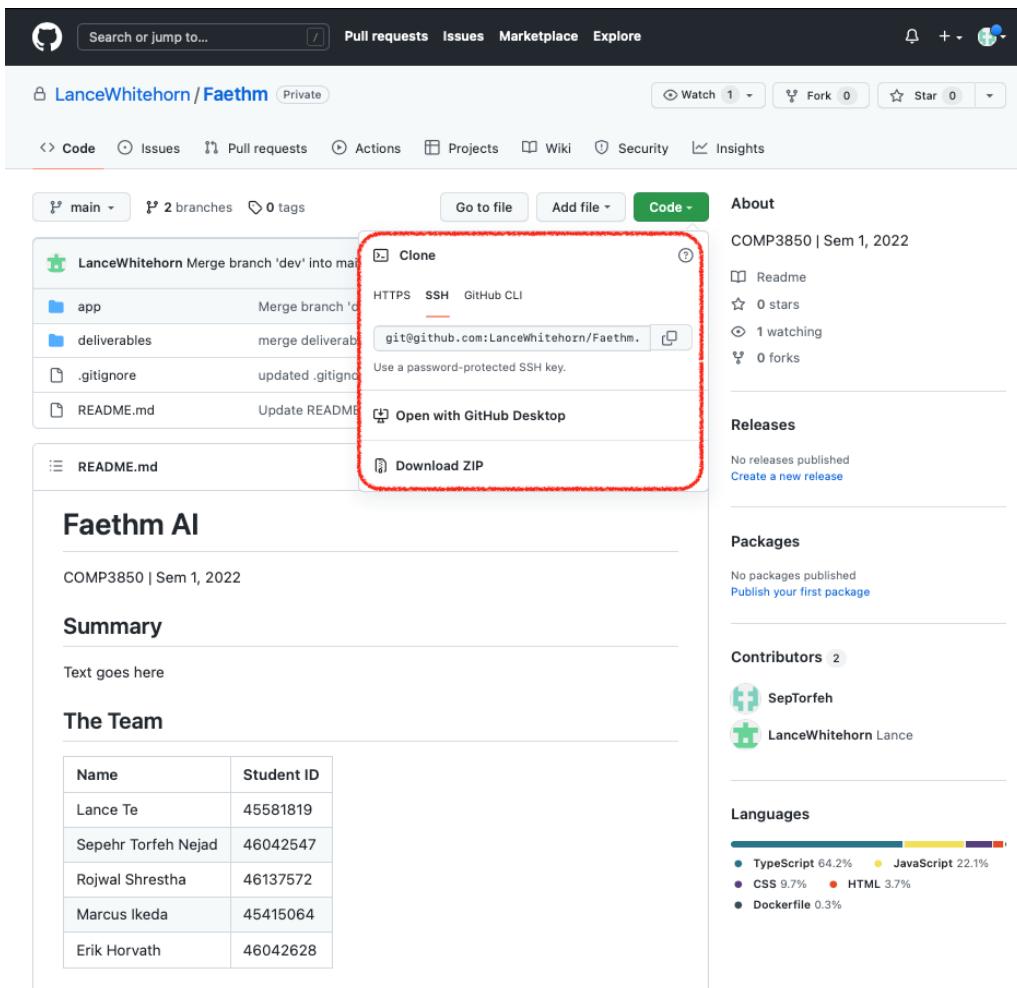
The Onboarding UI is a service that allows users to receive relevant tools and information in maintaining skills for work based on their personality and preferences on the main dashboard of Faethm platform. These preferences are obtained through a survey and are uploaded into Faethm's database which will then proceed to obtain tools and articles relevant to the user based on the results of the survey.

The storage solution that is used for the application is MongoDB. The remote MongoDB server will hold the information collected from the users. The code for the application will be hosted in a GitHub repository with further documentation of the software.

# 2. Installation Guide

## 2.1 Clone GitHub Repository

1. Our GitHub repository is private and the admin/developer responsible for installing the Onboarding UI should raise a request to have access to the repo.
2. Once user has access to the repo, they can clone the repository.



## 2.2 Installing Dependencies

### 2.2.1 Backend

User should navigate to “app” directory inside the root directory from the terminal and run command:

```
npm install
```

### 2.2.2 Frontend

User should navigate to “client” directory inside the “app” directory from the terminal and run command:

```
npm install
```

## 2.3 Seed Database

The following commands will seed / destroy the database with some sample users, the pre-defined Personas, and the pre-defined Use-Cases:

```
# Import data  
npm run data:import  
  
# Destroy data  
npm run data:destroy
```

## 2.4 Run Application

The following commands will run the application:

```
# Run frontend (:3000) & backend (:3001)  
npm run dev  
  
# Run backend only (with nodemon)  
npm run server  
  
# Run frontend only  
npm run client
```

## 3. Configuration Settings

The Onboarding UI uses the “dotenv” package for storing the app configuration which is based on Twelve-Factor app methodology. To configure the app, user should create a “.env” file in the root directory and the following:

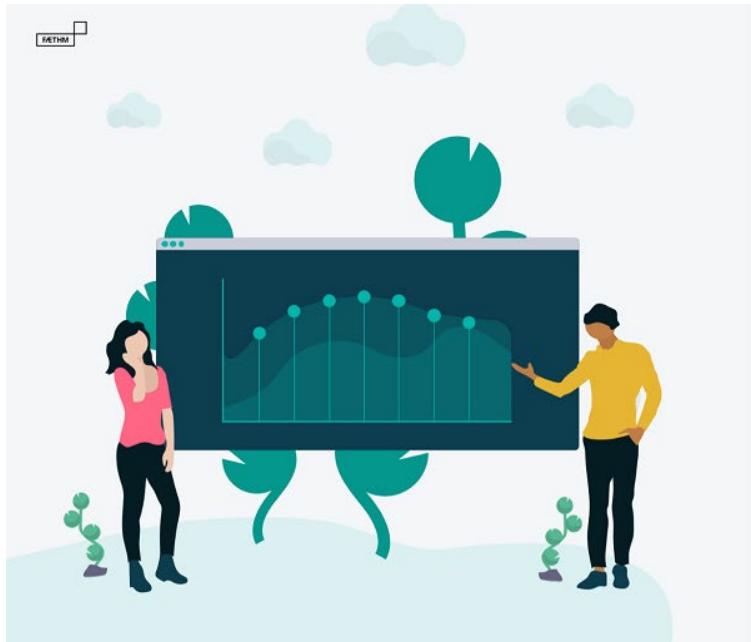
```
NODE_ENV = development  
PORT = 3001  
MONGO_URL = your mongodb uri  
JWT_SECRET = set your secret key for JWT token
```

## 4. Usage

### 4.1 End-User Usage

In this end-user documentation, we have provided some screenshots of the application to show the functionality of the software and its features.

#### 4.1.1 Login Page



The image shows a stylized illustration of two people, a man in a yellow shirt and a woman in a pink shirt, standing in front of a large digital screen. The screen displays a line graph with several data points. The background is a light blue sky with white clouds and green trees.

Welcome  
Log in to your Faethm account

Email address  
1 Type in email address

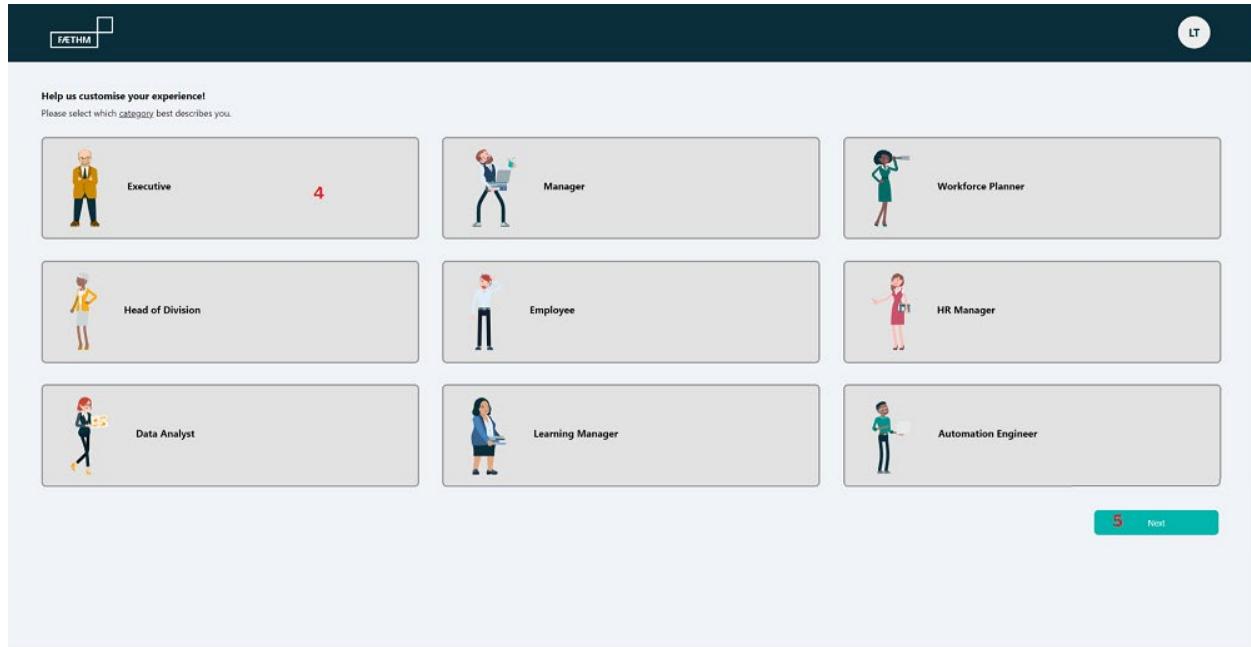
Password  
2 Type in password

3 Submit

The login page is presented upon accessing the application and will prompt the user to login to the application.

1. Field to enter an email associated with the user's Faethm account.
2. Field for password that is associated
3. Submits the information in the previous fields and will log the user in when provided a valid email and password.

#### 4.1.2 Persona Page



Upon logging into the application, if it is the user's first time using the service, they will be presented with the persona page, where they can choose their selected persona and a survey to customize their experience.

4. Tiles representing each of the available personas that can be chosen to customize the user's experience. Each tile is designed as a flip card and when hover over, use can see the description of the persona.
5. Proceeds to the survey page after a persona has been selected.

#### 4.1.3 Survey Page

The screenshot shows a survey page with four questions. Each question has three horizontal sliders labeled 'Low', 'Medium', and 'High'. The first question, '1. To what extent do you plan to maintain and upskill your current workforce?', has the 'Medium' slider highlighted with a darker shade. The other three questions are identical: '2. To what extent is your workforce willing to learn new skills and undertake further studies?', '3. To what extent do you believe your industry will be impacted by emerging technologies?', and '4. To what extent do you think your industry will be impacted by automation?'. A green 'Submit' button is located at the bottom right.

6. The darker boxes respond to the user's input upon selection and only one box in each row can be selected at a time.
7. Saves and submits the user's responses to Faethm's database, and moves the user to the main dashboard.

## 4.2 Admin / Developer Usage

The storage solution that is being used for this application is MongoDB which stores the collections for Personas, Use cases and Users. The Personas and Use cases are both relational to the Users since a user is needed to have an assigned persona and use case. Below is a screenshot of the Mongo Atlas dashboard when viewing the database.

Database Overview Real Time Metrics Collections Search Profiler Performance Advisor Online Archive Cmd Line Tools

DATABASE SIZE: 17KB INDEX SIZE: 11KB TOTAL COLLECTIONS: 3

Collection Name	Documents	Documents Size	Documents Avg	Indexes	Index Size	Index Avg
personas	10	541B	55B	1	36KB	36KB
use_cases	5	235B	47B	1	4KB	4KB
users	2	420B	210B	2	72KB	36KB

## 5. Further Assistance

For further assistance, the Faethm team can reach the team via the private Slack channel we have used to correspond throughout the course of this PACE Project.

The Faethm team can also reach the developers via email:

<b>Sepehr Torfeh Nejad</b> sepehr.torfehnejad@students.mq.edu.au	<b>Lance Te</b> lance.te@students.mq.edu.au
---	--

As the final deliverable is due in Week 16, the team will reasonably provide support for up to three weeks beyond Week 13.