

# SE 3XA3: Software Requirements Specification Euneva

Team 9, Euneva  
Mehta, Jash - mehtaj8  
Sharma, Aditya - shara24  
Ren, Zackary - renx11

February 12, 2021

# Contents

<b>1</b>	<b>Project Drivers</b>	<b>1</b>
1.1	The Purpose of the Project . . . . .	1
1.2	The Stakeholders . . . . .	1
1.2.1	The Client . . . . .	1
1.2.2	The Customers . . . . .	1
1.2.3	Other Stakeholders . . . . .	1
1.3	Mandated Constraints . . . . .	2
1.3.1	Product Constraints . . . . .	2
1.3.2	User Constraints . . . . .	2
1.3.3	Time Constraints . . . . .	2
1.3.4	Budget Constraints . . . . .	2
1.4	Naming Conventions and Terminology . . . . .	2
1.5	Relevant Facts and Assumptions . . . . .	3
1.5.1	Facts . . . . .	3
1.5.2	Assumptions . . . . .	3
<b>2</b>	<b>Functional Requirements</b>	<b>3</b>
2.1	The Scope of the Work and the Product . . . . .	3
2.1.1	The Context of the Work . . . . .	5
2.1.2	Work Partitioning . . . . .	6
2.1.3	Individual Product Use Cases . . . . .	7
2.2	Functional Requirements . . . . .	7
<b>3</b>	<b>Non-functional Requirements</b>	<b>9</b>
3.1	Look and Feel Requirements . . . . .	9
3.1.1	Appearance Requirements . . . . .	9
3.1.2	Style Requirements . . . . .	9
3.2	Usability and Humanity Requirements . . . . .	9
3.2.1	Ease of Use Requirements . . . . .	9
3.2.2	Personalization and Internationalization Requirements . . . . .	9
3.2.3	Learning Requirements . . . . .	10
3.2.4	Understandability and Politeness Requirements . . . . .	10
3.2.5	Accessibility Requirements . . . . .	10
3.3	Performance Requirements . . . . .	10
3.3.1	Speed and Latency Requirements . . . . .	10
3.3.2	Safety-Critical Requirements . . . . .	10

3.3.3	Precision and Accuracy Requirements . . . . .	10
3.3.4	Reliability and Availability Requirements . . . . .	11
3.3.5	Robustness or Fault-Tolerance Requirements . . . . .	11
3.3.6	Capacity Requirements . . . . .	11
3.3.7	Scalability or Extensibility Requirements . . . . .	11
3.3.8	Longevity Requirements . . . . .	11
3.4	Operational and Environmental Requirements . . . . .	11
3.4.1	Expected Physical Environment . . . . .	11
3.4.2	Requirements for Interfacing with Adjacent Systems . .	12
3.4.3	Productization Requirements . . . . .	12
3.4.4	Release Requirements . . . . .	12
3.5	Maintainability and Support Requirements . . . . .	12
3.5.1	Maintenance Requirements . . . . .	12
3.5.2	Supportability Requirements . . . . .	13
3.5.3	Adaptability Requirements . . . . .	13
3.6	Security Requirements . . . . .	13
3.6.1	Access Requirements . . . . .	13
3.6.2	Integrity Requirements . . . . .	13
3.6.3	Privacy Requirements . . . . .	13
3.6.4	Audit Requirements . . . . .	14
3.6.5	Immunity Requirements . . . . .	14
3.7	Cultural Requirements . . . . .	14
3.7.1	Cultural Requirements . . . . .	14
3.7.2	Political Requirements . . . . .	14
3.8	Legal Requirements . . . . .	14
3.8.1	Compliance Requirements . . . . .	14
3.8.2	Standards Requirements . . . . .	14
3.9	Health and Safety Requirements . . . . .	15
<b>4</b>	<b>Project Issues</b>	<b>15</b>
4.1	Open Issues . . . . .	15
4.2	Off-the-Shelf Solutions . . . . .	15
4.3	New Problems . . . . .	16
4.4	Tasks . . . . .	16
4.5	Migration to the New Product . . . . .	16
4.6	Risks . . . . .	16
4.7	Costs . . . . .	17
4.8	User Documentation and Training . . . . .	17

4.9	Waiting Room . . . . .	17
4.10	Ideas for Solutions . . . . .	17
<b>5</b>	<b>Appendix</b>	<b>18</b>
5.1	Symbolic Parameters . . . . .	18

## List of Tables

1	<b>Revision History</b> . . . . .	iii
2	Work Partitioning Part I. . . . .	6
3	Work Partitioning Part II. . . . .	6
4	Use Cases. . . . .	7
5	Use Cases Description. . . . .	7

## List of Figures

1	Context of Work for Euneva . . . . .	5
---	--------------------------------------	---

Table 1: **Revision History**

Date	Version	Notes
12/02/2021	1.0	Initial version of Requirements Specification Document

# **1 Project Drivers**

## **1.1 The Purpose of the Project**

Students may be managing five, six or even seven courses in one semester in addition to any extracurricular commitments they may have. Each course has an array of quizzes and assignments that students can easily lose track of due to the lack of consolidated information. The purpose of this project is to re-implement an existing To-Do list to allow students a way of accessing important dates and deadlines for courses on McMaster's University portal, Avenue. Balancing work, school and mental health is challenging, especially for younger individuals like students. Alleviating some of the stress from academics helps students avoid mental health issues and live a healthy lifestyle.

## **1.2 The Stakeholders**

### **1.2.1 The Client**

The clients of this product are the teaching staff of Software Engineering 3XA3 course that will be actively involved in the pre-development and development of the project. The teaching staff will be involved in the implementation decisions of the product, as well as determining if the project was successfully completed.

### **1.2.2 The Customers**

The customers of this product include students and professors attending/teaching at McMaster University. Students gain the ability to collate all assignments across all Avenue linked courses, as well as the ability to allow a better visual representation of upcoming dates and deadlines. Professors can use this product as a way to track student progress, as well as see if sufficient time has been allocated to each project and/or assignment.

### **1.2.3 Other Stakeholders**

Other stakeholders of this product include McMaster's Academic Success Centre, Brightspace (A2L Software), and Group-9 team members. The Academic Success Centre gains the ability to teach time management and pro-

ductivity tips. Brightspace, the Avenue to Learn software, gains from this product as it compliments the software they have developed.

## **1.3 Mandated Constraints**

### **1.3.1 Product Constraints**

In order to use the product, users will require an internet connection and a smartphone with either iOS or Android operating systems. Users will not need to remain connected to the internet once the data from Avenue has been collected, but will need to connect to the internet to allow for additional dates and deadlines to be added.

### **1.3.2 User Constraints**

Users of this product must have valid login information for Avenue to Learn to allow the product to access the dates and deadlines for each course.

### **1.3.3 Time Constraints**

The time frame given to complete this project and deploy the product is 3 months. This limits the amount of features and compromises product quality on the initial release of the software.

### **1.3.4 Budget Constraints**

The use of money in the development of this project is not permitted, this may hinder the user experience as well as the amount of features developed.

## **1.4 Naming Conventions and Terminology**

1. **UI** - User Interface
2. **UX** - User Experience
3. **To-Do** - Typical naming convention for a planner

## **1.5 Relevant Facts and Assumptions**

### **1.5.1 Facts**

1. The existing application has 1140 lines of code.
2. The existing application does not have web scraping functionality.
3. The existing UI contains the fundamentals of a To-Do list application, but needs additional features to fit our products usage.

### **1.5.2 Assumptions**

1. The application will be operated in an environment with a working internet connection for the initial setup, as well as for updating to-do items.
2. Users using the application for the intended purpose should have valid login credentials for Avenue to Learn.
3. Users are individuals currently attending or employed by McMaster University.

User characteristics should go under assumptions.

## **2 Functional Requirements**

### **2.1 The Scope of the Work and the Product**

The software product being developed in this document is a To-Do list linked to Avenue to Learn. The To-Do List will allow users with the correct credentials to extract dates and deadlines for assignments and upcoming tests from Avenue to Learn.

The goal of this software system is to provide the user, which in this case, students who attend McMaster University, with a way of organizing tasks, dates, and deadlines in a clearer manner.

The objective that this software system aims to achieve is to provide the user with ease of access to important information that they would require from

each course, in an organized but visually appealing manner.

The benefits of this software system is directly linked to student time management, and ease of access to important organized information. Providing the students with an easy way of accessing dates and deadlines relevant to their courses, would allow them to manage their time more efficiently.

**Deliverables:**

- Requirements Documentation
- Functioning Software

**Deadlines:**

- Requirements Document Rev 0 - 2021/02/12
- Proof of Concept Demonstration - 2021/02/22
- Test Plan Rev 0 - 2021/03/05
- Design Documentation - 2021/03/18
- Revision 0 Demonstration - 2021/03/22
- Final Demonstration (Rev 1) - 2021/04/05
- Final Documentation (Rev 1) - 2021/04/12



### 2.1.1 The Context of the Work

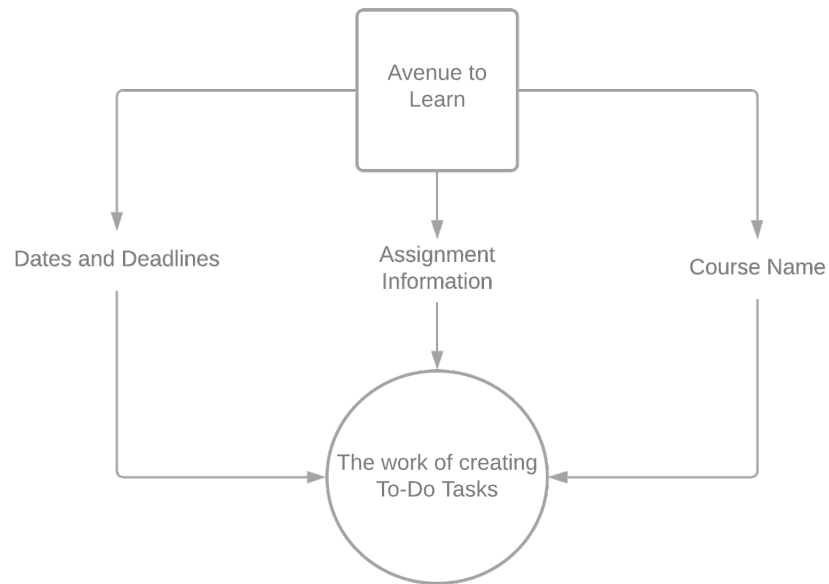


Figure 1: Context of Work for Euneva

### 2.1.2 Work Partitioning

Table 2: Work Partitioning Part I.

Event Number	Event Name	Input	Output
1	Gather Dates and Deadlines	Avenue to Learn Website	Dates and Deadlines available to use in to-do list
2	Gather Assignment Information	Avenue to Learn Website	Assignment Information available to use with corresponding dates and deadlines
3	Gather Course Names	Avenue to Learn Website	Course Names available to use with corresponding dates and deadlines.

Table 3: Work Partitioning Part II.

Event Number	Summary of Event
1	The dates and deadlines for a particular student are found in their account on Avenue to Learn. This allows the usage of dates for our implementation.
2	The Assignment Information collected will be used to attach corresponding dates to allow a more organized list of items.
3	The Course names are required for each date and deadline gathered for the same student. This allows for dates to be attached to a name.

### 2.1.3 Individual Product Use Cases

Table 4: Use Cases.

Actor	Actor's Goals	Use Case Name
Student	User links Avenue to Learn account with application.	Login (UC1)
Student	User selects courses to be added to the To-Do list.	Select (UC2)
Student	User edits information of added items from Avenue.	Edit (UC3)
Student	User changes dates of added items.	Edit (UC3)
Student	User deletes items from To-Do list.	Delete (UC4)
Student	User changes state of added To-Do item.	Edit (UC3)

Table 5: Use Cases Description.

Use Case	Description
Login (UC1)	The user intends to provide login credentials to access Avenue to Learn
Select (UC2)	The user intends to select the preferred courses to add to the To-Do list
Edit (UC3)	The user intends to edit fields in each To-Do item.
Delete (UC4)	The user intends to delete the selected To-Do item from the list.

## 2.2 Functional Requirements

FR1. The user interface must include a button for the user to login to Avenue to Learn. **Fit Criterion:** The login button will be displayed on the top right corner of the screen to allow ease of access to login to Avenue to

Learn.

FR2. The user interface must include an add task button. **Fit Criterion:** The button will be displayed at the bottom right corner of the screen to allow ease of access to add a task

FR3. The user interface must display the date selected. **Fit Criterion:** The date will be displayed on the top left corner of the screen to allow the user to view what date the current To-Do Items are located on.

FR4. The user interface must include a date selection at the top of the screen. **Fit Criterion:** The date selection will be displayed under the selected date to allow users to select a specific date.

FR5. The user interface must include a way to go to the today date. **Fit Criterion:** The displayed selected date on the top left corner must be a button, when pressed sets the date to today.

FR6. The user interface must display a dialog screen after the add task button is pressed. **Fit Criterion:** The displayed dialog should include “Task Name”, “Date”, “Time”, and a “Confirm” button to allow users to input individual tasks.

FR7. The user interface must display tasks under the date selection section. **Fit Criterion:** The displayed tasks must be visible and in its own section to allow better visibility.

FR8. The user interface must allow users to tap on a task to set it to done or cancelled. **Fit Criterion:** The user must be able to change each tasks individual status from to-do to done or cancelled.

FR9. The user interface must allow the user to swipe left on a task to edit. **Fit Criterion:** The user interface must display a button once a task is swiped left, to allow the user to edit the task using the same dialog screen used to add a new task.

FR10. The user interface must allow the user to swipe right on a task to delete. **Fit Criterion:** The user interface must display a button once a task

is swiped right, to allow the user to delete that specific task.

FR11. Once logged in to Avenue to Learn, the user interface must display all tasks and assignments. **Fit Criterion:** The user interface must display all tasks and assignments in the correct date and time corresponding to each task and assignment.

### 3 Non-functional Requirements

#### 3.1 Look and Feel Requirements

##### 3.1.1 Appearance Requirements

LFA1. The product shall have a user-friendly and aesthetically pleasing interface. **Fit Criterion:** The interface should distinguish each separate section to allow the user a better visual understanding of the product.

##### 3.1.2 Style Requirements

LFS1. The product should provide the user with a calm and distinguished experience. **Fit Criterion:** The user interface should use calm colouring as well as different colours for each section.

#### 3.2 Usability and Humanity Requirements

##### 3.2.1 Ease of Use Requirements

UHE1. The product shall be easy to use for students who have basic knowledge of smartphone applications. **Fit Criterion:** The user should be able to understand how the application works with a simple glance at the user interface.

##### 3.2.2 Personalization and Internationalization Requirements

N/A

### 3.2.3 Learning Requirements

UHP1. The product shall present an on-boarding to guide users through the application flow. **Fit Criterion:** The user can comfortably navigate around the product without assistance.

### 3.2.4 Understandability and Politeness Requirements

UHU1. The product shall use simple terms to describe actions and rules within the application. **Fit Criterion:** The user will be able to understand the terms the product is using.

### 3.2.5 Accessibility Requirements

UHA1. Colour is used to distinguish the different sections in the application, but colour-blindness will not affect the users ability to use the application. **Fit Criterion:** The user will be able to freely use the application even if affected by colour blindness.

## 3.3 Performance Requirements

### 3.3.1 Speed and Latency Requirements

PSL1. The product shall respond to the users actions and interactions within RESPONSE\_TIME. **Fit Criterion:** The application will respond to actions done by the user in the given RESPONSE\_TIME. Refer to Symbolic Parameters in Appendix for RESPONSE\_TIME.

### 3.3.2 Safety-Critical Requirements

N/A

### 3.3.3 Precision and Accuracy Requirements

PPA1. The dates, time, and names of each task will correspond to the task labeled in Avenue for every course. **Fit Criterion:** The tasks will be placed in the exact date and time that the deadline for an assignment is set to, as well as keep the name of the corresponding assignment.

### 3.3.4 Reliability and Availability Requirements

PRA1. The application can be accessed through a mobile device at all times.

**Fit Criterion:** The user will have access to the information in the application at all times.

PRA2. The application will update and add tasks when connected to the internet and logged into Avenue to Learn. **Fit Criterion:** The user will be able to update all tasks when logged into Avenue to Learn.

### 3.3.5 Robustness or Fault-Tolerance Requirements

PRF1. An invalid input will raise an exception. **Fit Criterion:** The user will see a message when an invalid input is used.

### 3.3.6 Capacity Requirements

PCR1. The performance of the product shall not be affected when high traffic of users are on the application. **Fit Criterion:** The application should be able to handle NUMBER\_OF\_USERS users at a time.

### 3.3.7 Scalability or Extensibility Requirements

PSE1. The product shall be capable of adding additional features without affecting the existing system. **Fit Criterion:** The code will be modular so new features will not affect the previous code.

### 3.3.8 Longevity Requirements

N/A

## 3.4 Operational and Environmental Requirements

### 3.4.1 Expected Physical Environment

EPE1. The software product will exist as an mobile application used specifically on mobile devices capable of connecting to the internet. **Fit Criterion:** The product will connect to the server via internet and will only be accessible via downloadable mobile packages.

### 3.4.2 Requirements for Interfacing with Adjacent Systems

RIAS1. The product shall be used on all android and iOS devices. **Fit Criterion:** The intended interface for the product shall be accessible by both android and iOS operating systems

### 3.4.3 Productization Requirements

N/A

### 3.4.4 Release Requirements

RR1. The product shall be released without any bugs or issues. **Fit Criterion:** The product will only be released when all test cases have been passed and all product use-cases are followed as expected.

RR2. The product shall be available to both iOS and Android users at the same time. **Fit Criterion:** The product will be compiled into a singular package/app manifest downloadable from our GitLab.

## 3.5 Maintainability and Support Requirements

### 3.5.1 Maintenance Requirements

MSM1. The product will undergo maintenance tests on a monthly basis. **Fit Criterion:** Test cases for the product will run periodically every month. Failed tests shall result in a notification to the developers.

MSM2. The product shall include contact information for the customers to contact if they encounter any problems. **Fit Criterion:** The product shall include contact details at the bottom of the page.

MSM3. The product shall be written in such a fashion that updates, bug fixes and regular maintenance checks can be performed/resolved easily. **Fit Criterion:** Tools like Doxygen, Prettier, ESLint and other formatting tools will be used to ensure proper formatting and readability.



### 3.5.2 Supportability Requirements

SR1. Emergency support shall be made available to the customer via email.

**Fit Criterion:** All pages/sections of the product shall include contact information for users to report potential issues or file inquiries.

### 3.5.3 Adaptability Requirements

AR1. The product shall be created such that it can run in any iOS and Android environment. **Fit Criterion:** A singular codebase will be used for the product with a cross-platform language.

## 3.6 Security Requirements

### 3.6.1 Access Requirements

AR1. All customers possessing an iOS and Android device enrolled within McMaster University shall be able to use and access the product. **Fit Criterion:** The product will require a McMaster email address, student ID and password in order to login and access the product.

AR2. Only members of the products' development team shall have access to the codebase. **Fit Criterion:** The product's source code will be stored on a private repo on GitLab

### 3.6.2 Integrity Requirements

IR1. The integrity of the product shall be protected by limiting access to the product, shall be error free and available for use 24/7. **Fit Criterion:** Only McMaster students will be able to use the product and will only be able to use the product when all bugs are resolved, all tests are passed and a successful deployment on docker is completed.

### 3.6.3 Privacy Requirements

PR1. The privacy of the customers shall be protected. **Fit Criterion:** The product will store valuable information in a secure database

PR2. The product shall not invade the customer's privacy more than what is required to successfully complete the tasks required of the product. **Fit**

**Criterion:** The product will only require a McMaster Email and the corresponding password for the user to use the platform.

#### **3.6.4 Audit Requirements**

N/A

#### **3.6.5 Immunity Requirements**

N/A

### **3.7 Cultural Requirements**

#### **3.7.1 Cultural Requirements**

CPC1. The product shall display text in the English language. **Fit Criterion:** All displayed code will use text in the English language.

CPC2. The product shall not display text or graphics that may be deemed offensive to religious or ethnic groups. **Fit Criterion:** The text and graphics displayed in the product will undergo a thorough review process before being used.

#### **3.7.2 Political Requirements**

N/A

### **3.8 Legal Requirements**

#### **3.8.1 Compliance Requirements**

N/A

#### **3.8.2 Standards Requirements**

N/A

### 3.9 Health and Safety Requirements

This section is not in the original Volere template, but health and safety are issues that should be considered for every engineering project.

N/A

## 4 Project Issues

### 4.1 Open Issues

- Existing Bugs
  - The existing bugs within the codebase need to be resolved before we can proceed with our own development.
- Different Coding Style and Quality
  - Standardized coding styles and automated formatters need to be run in order to format the codebase and identify inconsistencies so that they can be rectified.
- Lack of testing
  - Components that the existing codebase has introduced will need to be tested so that when we begin building on those components, further errors are not introduced and the program's behaviour is controlled

### 4.2 Off-the-Shelf Solutions

There are a variety of existing To-Do Lists existing on the market. These will all be competitors to our product. However, a key distinction is that our product will directly interface with Avenue to learn and is specifically for McMaster students. Below are examples of some potential competitors:

<https://www.notion.so/>

<https://todoist.com/>

### 4.3 New Problems

- Revamping Front-end
  - Since the original product was developed by another individual, it could be problematic to understand their code and their logic especially since there is no documentation about their product. As a result, it could prove to be challenging to complete the front-end related tasks we laid out in this specification.
- Developing Back-end
  - This is expected to be a challenge because of the premise of what we are attempting to do. We are attempting to scrape a highly secured website (Avenue). As a result, it could be challenging getting the authorization and permissions for a server to access Avenue periodically.

### 4.4 Tasks

The project plan that is available as a Gantt Chart is used for project planning and allocation of time for specific tasks.

### 4.5 Migration to the New Product

No phased implementation will be used to install the new system. Once the Avenue linked To-Do list is available to deploy, it will override the previous system that is in place.

### 4.6 Risks

Possible risks during product development and testing include:

- Difficulty Testing
- Improper Git Usage
- Bad Programming Practices
- Mismatch in User ID

## **4.7 Costs**

There are no costs for the application as it is all open source.

## **4.8 User Documentation and Training**

There will be no explicit user documentation. The user will be presented with an on-boarding guide upon the first launch of the product which can be revisited at any time.

## **4.9 Waiting Room**

Simplify UX for more efficient flow through the product.  
Implementation of a reminder notification system.  
Allow user to assign tasks a priority.

## **4.10 Ideas for Solutions**

Perform UI/UX development using external design tools, like Figma, before changing the code base.

# **References**

## 5 Appendix

This section has been added to the Volere template. This is where you can place additional information.

### 5.1 Symbolic Parameters

The definition of the requirements will likely call for SYMBOLIC\_CONSTANTS. Their values are defined in this section for easy maintenance.

SP1. **RESPONSE\_TIME**: 2 seconds

SP2. **NUMBER\_OF\_USERS**: 50