Requirements Document University Management Software

Psychic Octo Guacamole

March 2nd, 2023

Table of Contents

Revision History

Name	Date	Reason for Changes	Version
All	01/26/2023	Skeleton Document	0.0
All	01/29/2023	1, 2, 3, 4, 5 Drafting	0.1.0
All	01/29/2023	Final Drafting	0.2.0
All	01/29/2023	Final Copy	1.0
All	03/02/2023	Changes made based on Request for Changes Received	1.1

1 Introduction

1.1 Purpose

Requirements Document (RD) describes the functionality of a University Management Software (UMS) that handles the financial data of the students and academic staff for the University of Victoria (UVic). UVic's current system does not meet the modern standards for ease of use and accessibility while also lacking features such as analytics and reminders. This RD is meant to cover the requirements laid out in UVic's Request for Proposal (RFP) and create a module that is easily able to integrate into the University's current processes.

1.2 Project Scope

The project's purpose is to redesign, streamline, and replace the current financial and accounting management system for students. This new system will provide options for students at UVic for budgeting, accounting, and reporting. Team University of Victoria's goal is to fully replace their systems since they are inefficient and cause student confusion when trying to pay their tuition. This leads to a worse student experience which is bad for their business as it requires more technical assistance employees to be hired.

1.3 Glossary of Terms

Term	**Definition**
SSO	Stands for "Single Sign On", it allows a user to log in from a single ID from several software systems.
Tuition	The amount of money a student owes for their registration in courses at UVic.
The System	The final product to be developed by Psychic Octo Guacamole.
Term	A series of months (i.e. Winter, Fall, Summer) that students take courses within, AKA semester.
API	Stands for "Application Program Interface". The layer that transfers information between the database and the website.
Percentage System Uptime	The percent of time that the system is guaranteed to be up working reliably.
OneCard	A card UVic students can use to make payments.

1.4 References

[1] J. Demian, “Top 10 most important metrics to measure website performance,” Sematext, 23-Mar-2020. [Online]. Available: https://sematext.com/blog/website-performance-metrics. [Accessed: 29-Jan-2023].

1.5 Overview

The RD 1.0 starts by describing an overall description of the product. This includes, the product perspective, product features, user characteristics, operating environment, design and implementation constraints, and assumptions and dependencies. Furthermore, this document will describe the system's features, that include a description, the priority and functional requirements of each one. After that, the document will provide the External Interface Requirements for the user, hardware, software and communications interface. The next section contains Other Non-Functional Requires such as Performance, Security, and Software Quality Attributes. Lastly, the document contains an Appendix that contains a list of issues.

2 Overall Description

2.1 Product Perspective

The university financial system to be developed will be a replacement of the current system in place at the University of Victoria. The perspective for the proposed product is as follows:

- The product will replace the web-application that is currently being used which includes the following features:
    - Creating and logging into accounts to view student financial data
    - Viewing tuition owed per term
    - Being notified when a payment deadline is approaching
    - Sorting information by month and year
- The application will integrate with major Canadian banks and external payment processors, such as Visa credit cards, via a method TBD.
- The application will use a service TBD to send notifications to users.

2.2 Product Features

The goal of the product is to allow students to pay their tuition and view their financial data. The features this will include is allowing the students to: log in and out of the service, view their tuition term, setup and be notified when payments are due, budgeting system so that the student can project how much the term will cost.

2.3 User Classes and Characteristics

Students are the primary kind of user who will be using the system. They will be able to view their tuition accounts, get access to analytics on their own data, and get reminders on payment due dates. They will also be able to access budgeting tools to help control their UVic finances, especially those who have funds on their OneCard or have a meal plan.

2.4 Operating Environment

- The software will be accessible by computers and smartphones that have access to the internet.
- Hardware platform - The system will be hosted on an cloud service provider
- Browser applications - The web-application will be accessible by the major browsing applications (Mozilla Firefox, Safari, Chrome, and Edge).
- Operating System and Version - N/A

2.5 Design and Implementation Constraints

- Keeping design in line with UVic's general design language.
- Getting access to the financial data of Students whilst adhering to security and privacy laws.
- Learning and implementing UVic's course API

2.6 Assumptions and dependencies

2.6.1 Assumptions

Payment Currency

The service will show charges and past payments in CAD.

Types of Users

There will be two types of users. An administrative user to handle any errors or student problems that occur. Secondly, a student user that will be using the service provided.

User Accounts

Since this is a UVic system, we will be doing authentication through UVic's Netlink service.

User Payments

The system will allow users to make payments towards UVic.

2.6.2 Dependencies

Third party services

Third party APIs will need to be used in order to support external payment processors, such as Visa.

UVic Registration System

The system will integrate and pull in student data in order to determine a student's balance for a given semester.

3 System Features

3.1 User Accounts

3.1.1 Description and Priority

Students interact with the management software using the Netlink accounts.

Priority: High

3.1.2 Functional Requirements

REQ-1-1: Users should be able to log into the system with their netlink ID, using UVic's SSO.

REQ-1-2: Users should be able to log out of their account.

REQ-1-3: Users should be able to manage notification settings.

3.2 View Financial Statements by Term

3.2.1 Description and Priority

Users are able to view their financial information organized by term. This will include information such as payments made, balance owed, total term charges, and a list of all charges and prices for each. Furthermore, the system will provide information regarding received scholarships and bursaries.

Priority: High

3.2.2 Functional Requirements

REQ-2-1: Users should be able to filter financial information by term.

REQ-2-2: Users should be able to view financial statements by term.

REQ-2-3: Users should be able to filter financial information by year.

REQ-2-4: Users should be able to filter financial information by month.

3.3 Make Payments

3.3.1 Description and Priority

After tuition is calculated, the student must be able to make payments through our system and update all systems associated with this payment. The services that must be updated after a payment is made are the database, all webapps connected to the student's account, and our notification system so that they get confirmation of their payment.

Priority: Medium

3.3.2 Functional Requirements

REQ-3-1: Users should be able to make a payment.

REQ-3-2: Users should receive a notification after making a payment.

REQ-3-3: Users should be able to view a breakdown of their due payments.

REQ-3-4: Users should be able to pick a saved payment method to pay with.

REQ-3-5: Users should be able to view deadlines for payments to be made.

3.4 Notifications

3.4.1 Description and Priority

When sensitive deadlines are approaching (last date of dropping, payment deadlines, etc.), the student will have settings be notified via email or web app notification.

Priority: High

3.4.2 Functional Requirements

REQ-4-1: Users should be able to choose their medium for notifications. They will be able to choose between receiving email notifications or text notifications.

REQ-4-2: Users should be able to choose which notifications they want to receive.

REQ-4-3: Users should be able to set when they receive notifications for a specific deadline or event.

REQ-4-4: Users should receive notifications through their medium (email or web app notification) chosen and when they specified.

REQ-4-5: Users should be able to unsubscribe from all notifications.

3.5 Reporting and Analytics

3.5.1 Description and Priority

User's are able to view reports and analytics regarding their financial information by term and year.

Priority: Low

3.5.2 Functional Requirements

REQ-5-1: Users can view reports and analytics of payments by year. Reports will include spending categories on their onecard, and tuition breakdown for how much of their tuition is spent on courses compared to other things.

REQ-5-2: Users can view reports and analytics of payments by term. Reports will include spending categories on their onecard, and tuition breakdown for how much of their tuition is spent on courses compared to other things.

REQ-5-3: Users can view their projected costs of schooling.

3.6 Budget Tools

3.6.1 Description and Priority

User's are able to set budgets for how much they are able to spend in a given time period on their OneCard.
Priority: Low

3.6.2 Functional Requirements

REQ-6-1: Users should be able to set a maximum spending target for a given term.

REQ-6-2: Users should be able to set a maximum spending target for a given year.

REQ-6-3: Users should be able to see daily spending targets based on their set budget.

REQ-6-4: Users should be able to configure notifications for when they are approaching spending goals.

4 External Interface Requirements

4.1 User Interfaces

The system will be a standard web page that is reactive and looks clean and is easy to use on all devices. The core of the interface will remain consistent across the different devices. The design will be responsive, meaning it is fully accessible on devices of all sizes. Furthermore, the interface will be fully accessible to allow everyone to be able access and use all features on the site. The main page will provide students with an overview of important financial details. Upcoming payments and notifications will be available here. The “payments” page will be accessible from the header where students can see all payments and continue to pay each one. Further UI choices will be considered after discussion with the clients.

4.2 Hardware Interfaces

The product is planned to be a web page that will be able to run on desktops, tablets, and other mobile devices. All utilities will be carried out by user touch via touch screens or through mouse clicks on desktop devices.

4.3 Software Interfaces

A user will be able to login using UVic’s authentication system. Once they are logged in, their UVic financial data will be synced automatically. From here, the user will be able to look at their payment history, get a breakdown of their tuition costs, and look at reports from other semesters.

4.4 Communications Interfaces

This product will need to communicate directly with UVic’s servers in a secure way to access a user’s relevant financial information. This will allow us to show the user their data in a simple and easy to understand way. The product will also need to connect to an email server to send out emails. These emails will be reminders that users will get notifying them that a new report has been created, or that they need to pay off their tuition balance.

5 Other Non Functional Requirements

5.1 Performance Requirements

The performance requirements include speed, uptime and error rate. The web-application must be responsive allowing for a seamless user experience when navigating the page and its features. The speed of the system should allow for this seamless experience by providing a load time of less than 3 seconds [1]. The system is important for students, therefore it must have an system uptime of 99.999% [1] throughout the year. Allowing students to access their financial information at any point. Lastly, as the system relates to finance, the system should have a lower than 0.001% error rate per 1000 students.

5.2 Security Requirements

Data from UVic's servers must be taken securely and not be permanently stored on any foreign servers. User's will be automatically logged out after 5-10 minutes of inactivity to secure their account and prevent private information from leaking or an option to stay logged in for 8 hours. Data storage and data transfer between interfaces must follow standard security protocols (data encryption in transit and at rest). The web application will not store any payment information.

5.3 Software Quality Attributes

Development will be done with modifiability, evolution and adaptability in mind. This will allow the product to be relevant today, whilst allowing it to easily and quickly be updated later on down the line. A lot of financial software can be confusing to understand on the first use. Our goal will be for our product to be easy to pick up and understand immediately. No user should be put off using the software because of confusion from the software's layout or functionality.

Appendix: Issues List

- How the system will integrate with the major Canadian banks has yet to be determined. There will be external API integration with systems that are currently unknown.
- A system that can send automated emails to users in the form of notifications has yet to be determined. A third party system may be needed to accomplish this.