Visual Course Planner

Design Document

Version 1.0
11.12.2018

Table Of Contents

1 INTRODUCTION

1.1 INTRODUCTION

Organizing a degree plan at the University of British Columbia is a tedious and confusing task with the current way that degree planning is organized. A static page for each program features a list of required courses to be taken to fully complete the program. A student may browse all courses on the Course Description page, however not all courses are guaranteed to be offered every year, or even at all.

The Visual Course Planner (VCP) is designed to assist students or academic advisors to plan an entire degree at the University of British Columbia Okanagan in one application.

The VCP is visually pleasing featuring a contemporary UI design and drag and drop controls. The VCP enables students to create an up-to-date degree plan, removing any stress of missing credits or empty credits (credits that do not count for their degree).

1.2 PURPOSE OF DOCUMENT

This document outlines the design and underlying structure of the Visual Course Planner, including the design constraints, system architecture, user interface, limitations and test plan.

1.3 MISSION STATEMENT

The most efficient way for planning your future at UBC Okanagan.

1.4 SYSTEM OVERVIEW

The Visual Course Planner (VCP) is a web application that allows students and academic advisors to plan degrees at the University of British Columbia Okanagan campus. The design of the Visual Course Planner is to enhance the students ability to successfully plan their degree at UBCO without having to worry about missing credits, and taking courses that don't apply to their degree.

The VCP features a total of four user interfaces: Login/Create account, Student Information, Admin Portal, and the planner interface. The planner interface will feature autosave on any changes the user makes. The VCP will support all major web browsers.

There are only two types of users (admins or students) who can create an account to interact with the VCP. Admins and Students login using the same login page.

Admins have the ability to upload a file containing the current offered courses and program requirements based on what the University is offering. Admins have the ability to upload more than one file at a time. Once the data has been uploaded it will then be displayed and accessible by each student when they are making a degree plan. Admins will not have any access to students' accounts. The information available to the user depends on what the admins upload. Unless an admin indicates a course is offered, students will not be able to add that course to their degree plan.

Users will be able to create an account by providing their personal information such as their first name, last name, email and their chosen password. After, an confirmation email will be sent to the email address that was provided. Once a student account is created and confirmed via the confirmation email, they will then be prompted to fill in additional information about themselves such as which degree they wish to enroll in, desired major, and previously completed courses. A degree *must* be selected in order for the student to proceed to the

next step. Whilst the student is deciding on a degree, a description of what the program entails is presented on the right hand side of the screen. The description presented is provided by UBC.

Once a student has provided their information, they will be taken to the Visual Course Planner interface. On the left side of the planner interface is a static vertical menu that displays the student's name, a logout button, a save plan button, a list of previously saved degree plans, a "favourite plan" button, a "new plan" button, and a note section. On the far right is a minizable menu that displays a search bar, present warnings based on a current change, and a description of a previously selected course. Across the top of the window is an optimize button, and a warning summary. The optimize button will shrink the students entire degree plan to the shortest duration possible based on which courses are included.

The main component of the VCP is located in the middle and features the courses which the student plans to take during their time at UBCO. Courses are organized by year and then by semesters. If summer courses, Go-Global and or Co-op courses are added to the degree plan, corresponding categories will be added (i.e. Summer Courses; Term 1, Term 2). Here, they can utilize drag and drop functionality to drag a courses into their plan depending on the term the course is offered. All required courses are outlined in a salmon pink colouring while the elective courses are outlined in electric blue. Due to the limitation of screen space, the area that displays all courses for a degree plan is scrollable rather than scalable to avoid shrinking all elements into the restricted area.

This web application will follow the MVC structure and use a REST API. The UI will make request JSON data through a rest call to the server. The JSON data depends on the type of event triggered. The server queries the database and builds a JSON object from the query results. Once the UI receives the JSON, it will update the view accordingly.

A data scraper will retrieve course data from the UBC Okanagan academic calendar. This data will be formatted and imported into the system.

For a detailed description on how the VCP is organized please see section 3: System Architecture.

1.5 USER GROUPS

1.5.1 TYPES OF USERS

1) Student
2) Admin
3) Individuals with an interest in the software

1.5.2 USE-CASE MODEL

Actor	**Goal**
Student	To securely sign up or login and create a visually pleasing course plan.
Admin	To maintain the VCP by uploading a file containing the most current and up-to-date offered courses at UBC Okanagan as well as current degree requirements.

Student Use Case

Name:	Log in
Flow of Events	1. User enters their email and password 2. User waits for server to verify credentials 3. If credentials are invalid, User is presented with login error 4. If credentials are valid, User is redirected to the Student Information Interface
Pre-Conditions	User has a device with an internet browser and an internet connection User is on the VCP tool on their internet browser User has an account User has established a connection with the server
Post-Conditions	If validated, user is successfully logged in and redirected to Student Information Interface If credentials are invalid, user can restart flow of events or exit

Name:	Create Plan
Flow of Events	1. User can select a new degree plan to create 2. User can select a default degree plan to edit or use
Pre-Conditions	User is logged into the system User has an ongoing connection with the server User knows what degree they are making a plan for (currently computer science)
Post-Conditions	User has created a plan for their degree visualization

Name:	Add/Remove Course
Flow of Events	1. User can add and or remove a course from their degree plan 2. The warning system checks for any warnings thrown 3. If a warning is thrown the user is informed 4. The user can fix their plan according to the type of warning thrown
Pre-Conditions	User is logged into the system User has an ongoing connection with the server User searches for the course they are looking for in the search bar User correctly spells the course name
Post-Conditions	User has added a course to their degree plan User can now visually see their course and how it fits in their degree plan

Name:	Move Around Courses (Drag and Drop)
Flow of Events	1. User can move courses around to freely plan their degree 2. Warning system checks for any warnings thrown when moving courses
Pre-Conditions	User is logged into the system

	User has an ongoing connection with the server User has browser capabilities for drag and drop features
Post-Conditions	If any warnings are thrown from course placement, user can adjust plan to correct the warning(s)

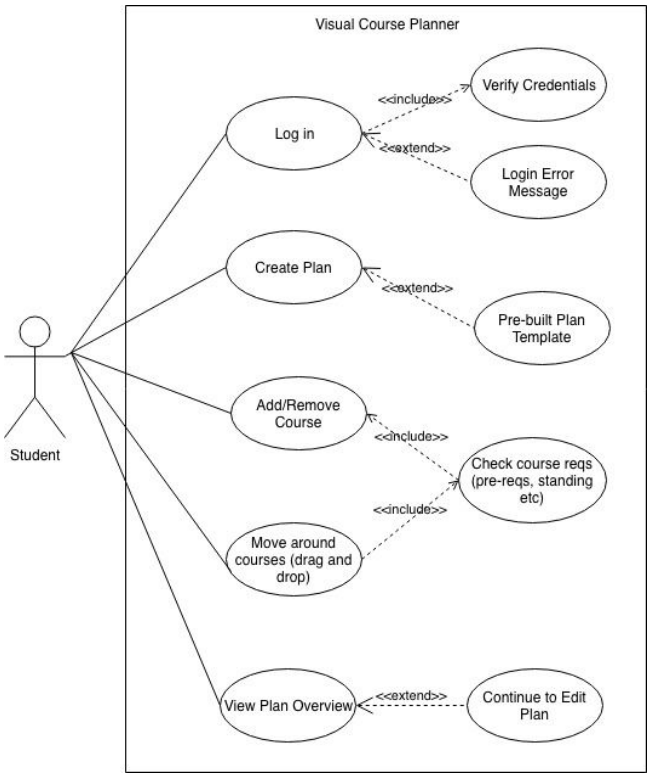
Name:	View Plan Overview
Flow of Events	1. User has completed their degree plan 2. User can continue to make edits to their plan if they wish 3. User can finalize their plan and make a new one if they wish
Pre-Conditions	User is logged into the system User has an ongoing connection with the server User has made sure to correct all warnings if any are present including pre-req, standing, degree requirement warning(s), etc
Post-Conditions	User has successfully created a degree plan

Admin Use Case

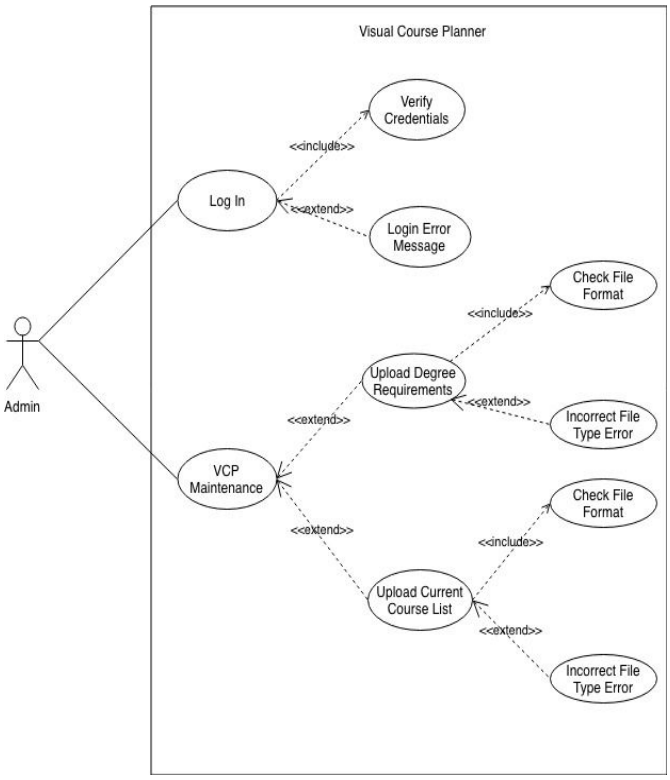
Name:	Log in
Flow of Events	1. Admin enters their login credentials 2. Admin waits for server to verify credentials 3. If credentials are invalid, admin is presented with login error 4. If credentials are valid, admin is redirected to the admin portal
Pre-Conditions	Admin has a device with an internet browser and an internet connection Admin is on the VCP tool on their internet browser Admin has an account Admin has established a connection with the server
Post-Conditions	If validated, admin is successfully logged in and redirected to the admin portal If credentials are invalid, admin can restart flow of events or exit

Name:	VCP Maintenance
Flow of Events	1. Admin can upload current course list and or degree requirement file(s) 2. Server checks for file format and type 3. If file is incorrect, warning is shown to admin
Pre-Conditions	Admin is logged into the system Admin has an ongoing connection with the server Admin has a correctly formatted file to upload
Post-Conditions	If no warnings thrown, admin has successfully uploaded VCP maintenance files

1.5.3 STUDENT USE-CASE DIAGRAM



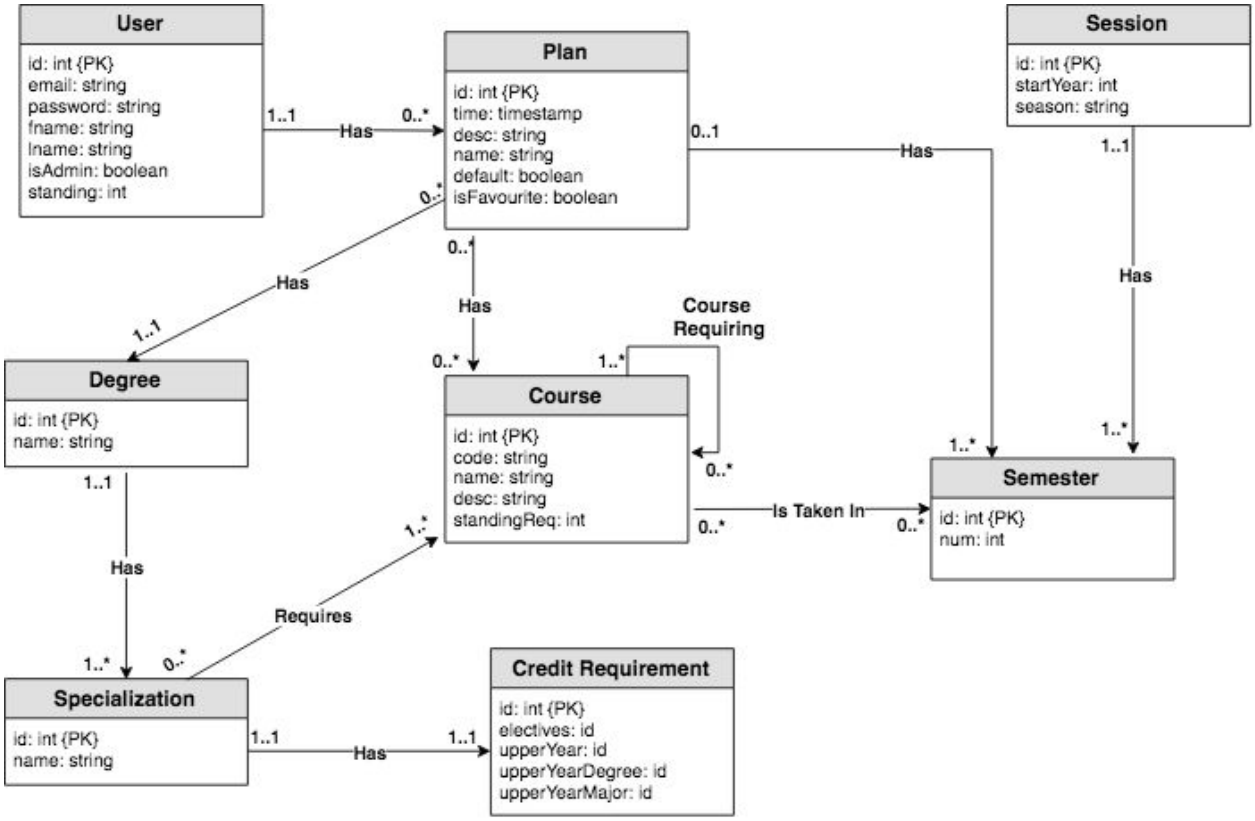
1.5.4 ADMIN USE-CASE DIAGRAM



2 SYSTEM ARCHITECTURE

This section describes the necessary architecture to achieve the Visual Course Planner's design.

2.1 ER DIAGRAM



2.1.1 ENTITY ATTRIBUTE DESCRIPTION

2.1.1.1 User

A student is a user that can either be a student of the University of British Columbia Okanagan or a student of interest. To successfully create an account, a first name, last name, email address and password must be provided. An email confirmation will be sent out to the provided email address to confirm a successful sign up. Upon a successful sign up, an unique ID will be assigned to the account.

ATTRIBUTE	**DESCRIPTION**
id: INT **[PK]**	The user's (admin or student) unique ID assigned to the account
email: VARCHAR(50)	The user's (admin or student) registered email address

pass: VARCHAR(50)	The user's (admin or student) password
fname: VARCHAR(50)	The user's (admin or student) first name
lname: VARCHAR(50)	The user's (admin or student) last name
isAdmin: BOOLEAN	Checks to see if account is an admin or student
standing: INT	The year level the student is currently in

2.1.1.2 Plan

Plan is assigned an unique ID. The Plan contains the time the plan was made, an optional description provided by the student, a name for the degree plan, also provided by the student, and a boolean variable that checks to see if the requested plan is a default plan.

ATTRIBUTE	**DESCRIPTION**
id: INT **[PK]**	The plan's unique ID
time: DATETIME	The date and time the plan was created
desc: VARCHAR(300)	The description of the plan (optional)
name: VARCHAR(50)	The name of the plan (optional)
default: BOOLEAN	Checks to see if plan is a default plan or one created by Student
isFavourite: BOOLEAN	Checks to see if plan is favourited or not

2.1.1.3 Course

The Course contains all the information about a specific course offered at UBC Okanagan. Each course Is assigned an unique ID. Code is the course code pre-assigned by UBC, and name and description are assigned to each course which is also identical to what is registered by the university.

ATTRIBUTE	**DESCRIPTION**
id: INT **[PK]**	The course's unique ID
code: VARCHAR(50)	The course code (pre-assigned by UBC)
name: VARCHAR(50)	The name of the of the course (pre-assigned by UBC)
desc: VARCHAR(50)	The description of the course (pre-assigned by UBC)
standingReq: INT	The year level a student must be in to take the course

2.1.1.4 Semester

Semester specifies which semester each course is offered in depending on the current year. An ID, the school year and number (range 1 -4) is assigned. Semester num 1 is Winter Term 1, semester num 2 is Winter Term 2, and semester num 3 and 4 are the two semesters for during Summer.

ATTRIBUTE	DESCRIPTION
id: INT	The semester's id
schoolYear: INT	The school year of the offered course in the specific semester
num: INT	Semester number

2.1.1.5 Session

Session is the winter session that includes a year and a season - something like "2018W".

ATTRIBUTE	DESCRIPTION
Id: INT	Unique id
startYear: INT	The year that the session started in
season: INT	The season that the session started in

2.1.1.6 Credit Requirements

The amount of credits a particular major requires in specific areas.

ATTRIBUTE	DESCRIPTION
Id: INT	Unique id
electives: INT	The number of elective credits needed
upperYear: INT	The number of upper year credits needed
upperYearDegree: INT	The number of upper year degree specific credits needed
upperYearMajor: INT	The number of upper year major specific credits needed

2.1.1.7 Degree

A degree that a user would be enrolled in.

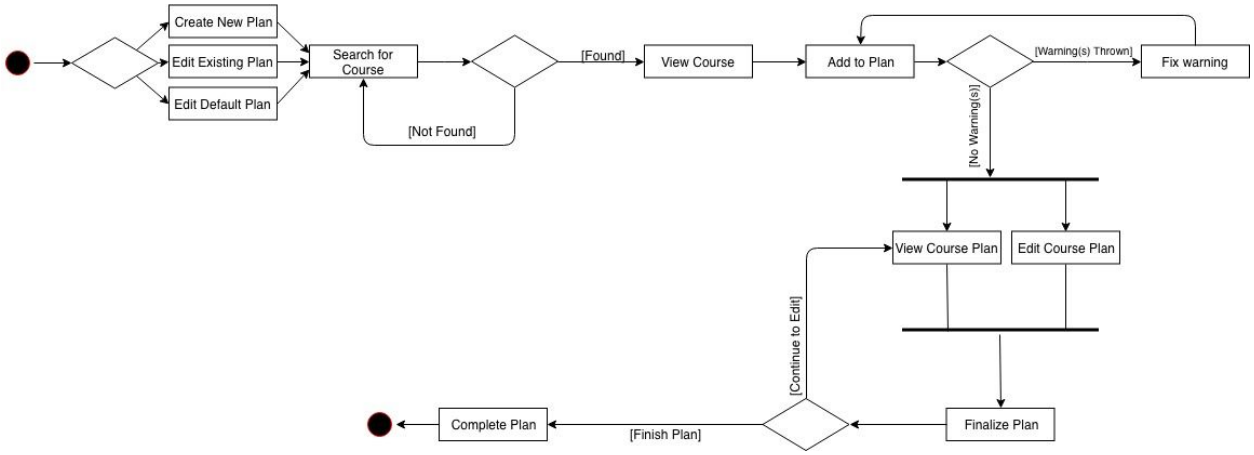
ATTRIBUTE	DESCRIPTION
Id: INT	Unique id
name: VARCHAR(100)	The name of the degree according to UBC

2.1.1.8 Major

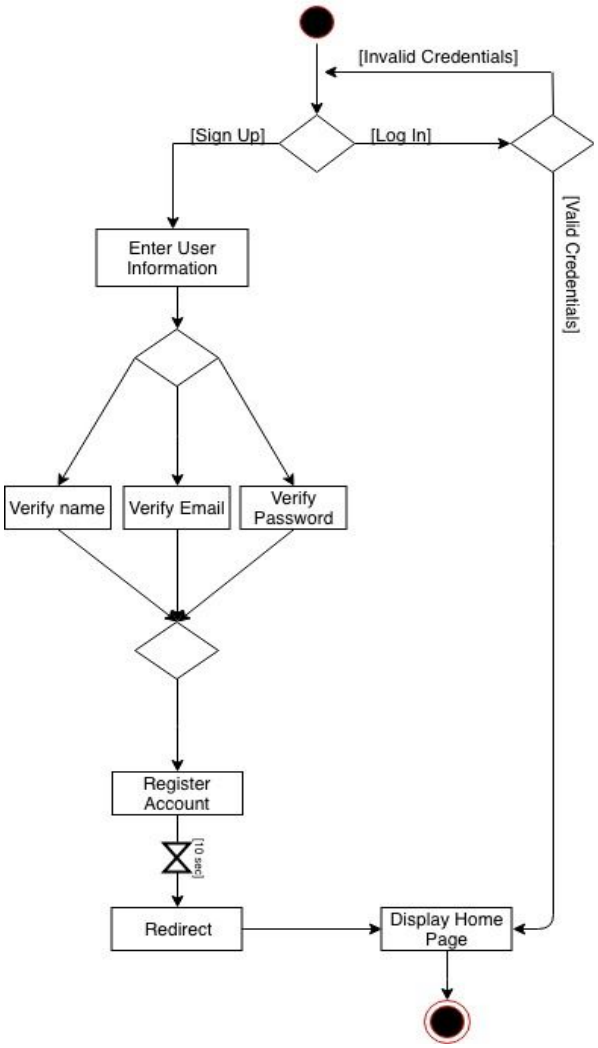
ATTRIBUTE	DESCRIPTION
Id: INT	Unique id
name: INT	The name of the major according to UBC

2.2 ACTIVITY DIAGRAMS

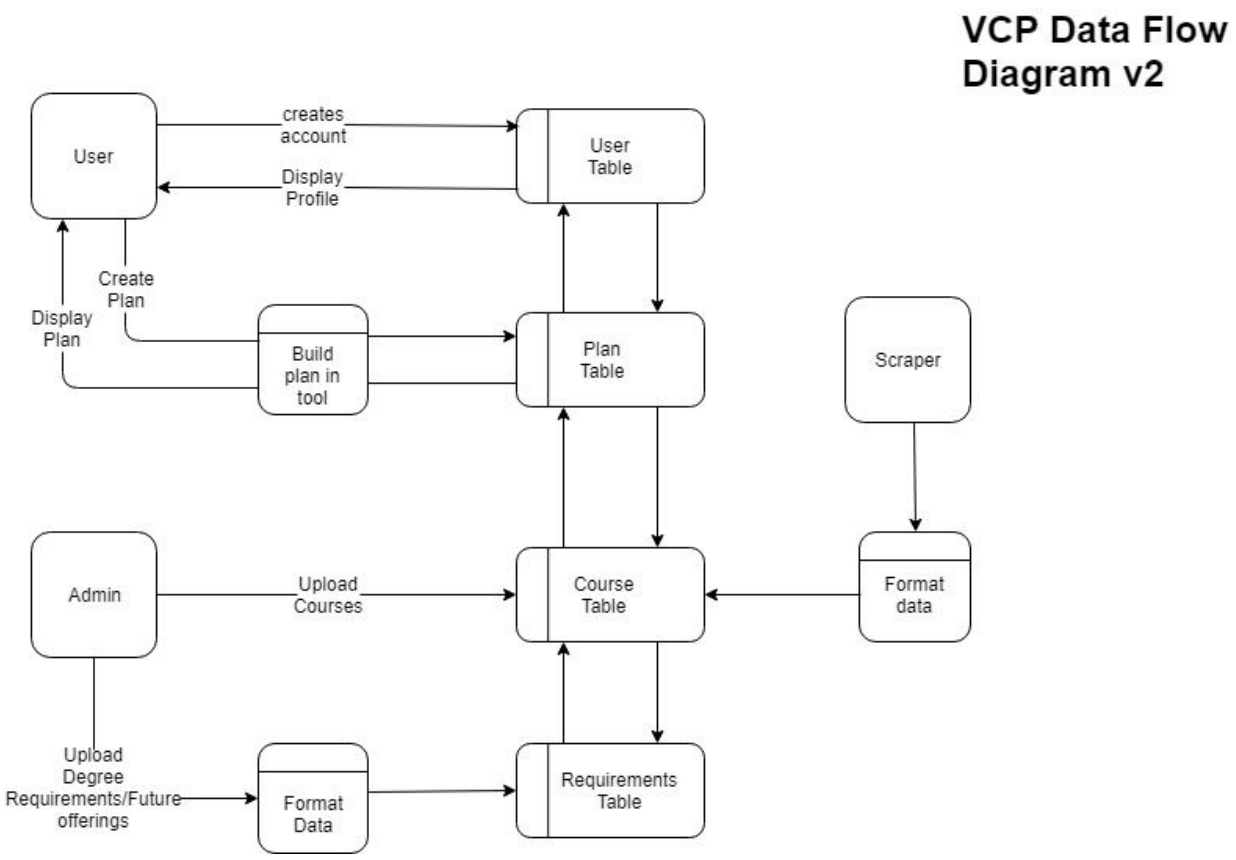
2.2.1 Use case: Create a plan



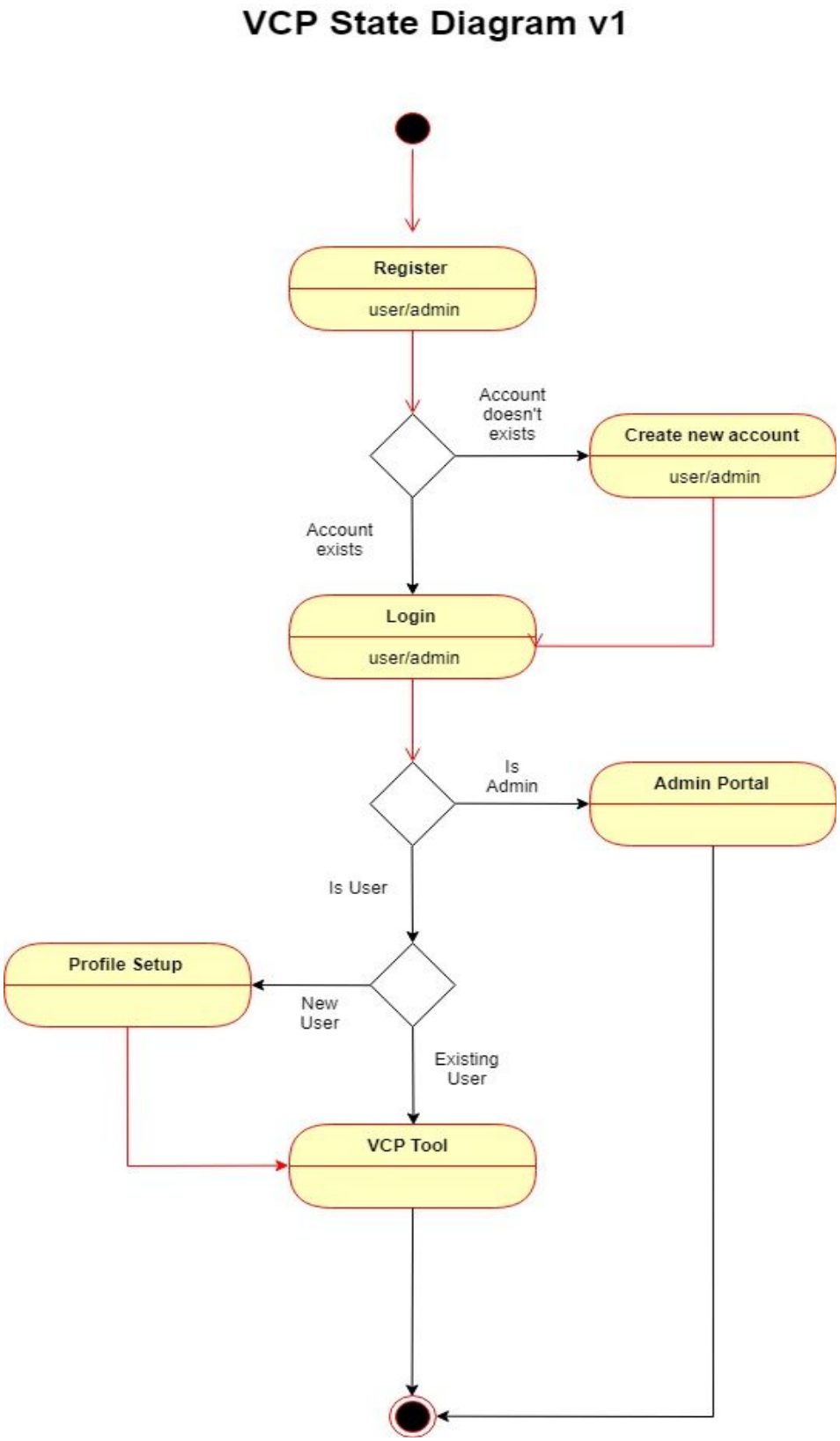
2.2.2 Use case: Signup / Login



2.3 DATA FLOW DIAGRAM

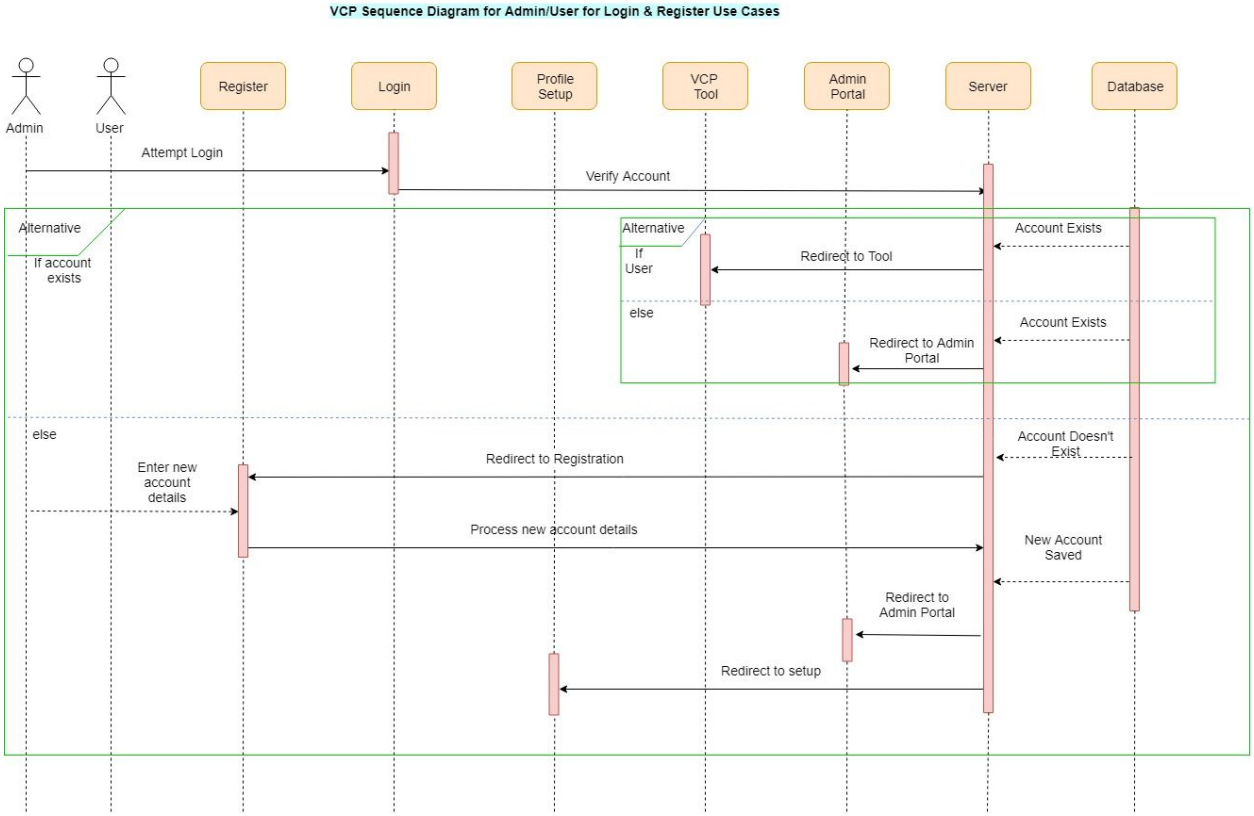


2.4 STATE DIAGRAM

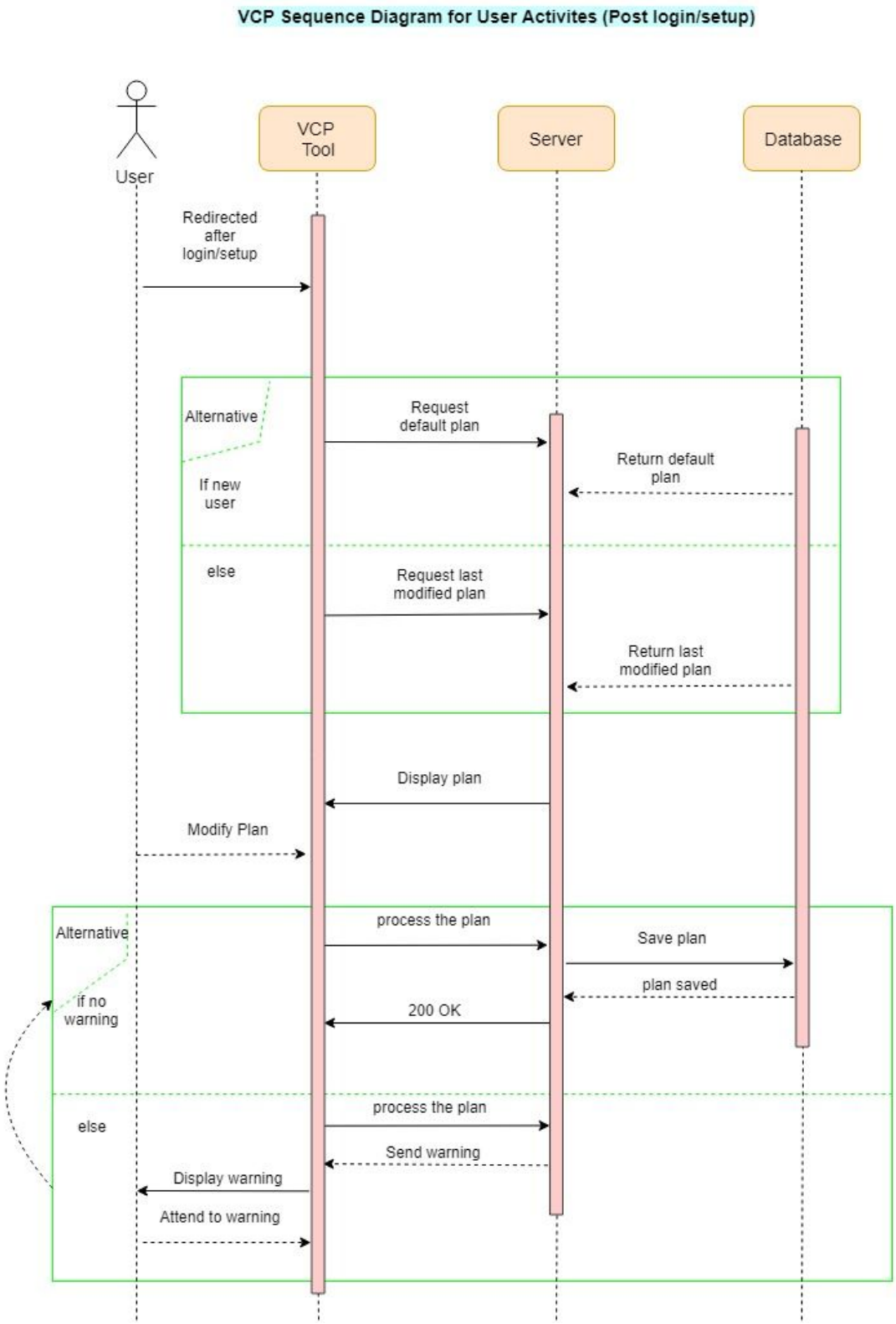


2.5 SEQUENCE DIAGRAMS

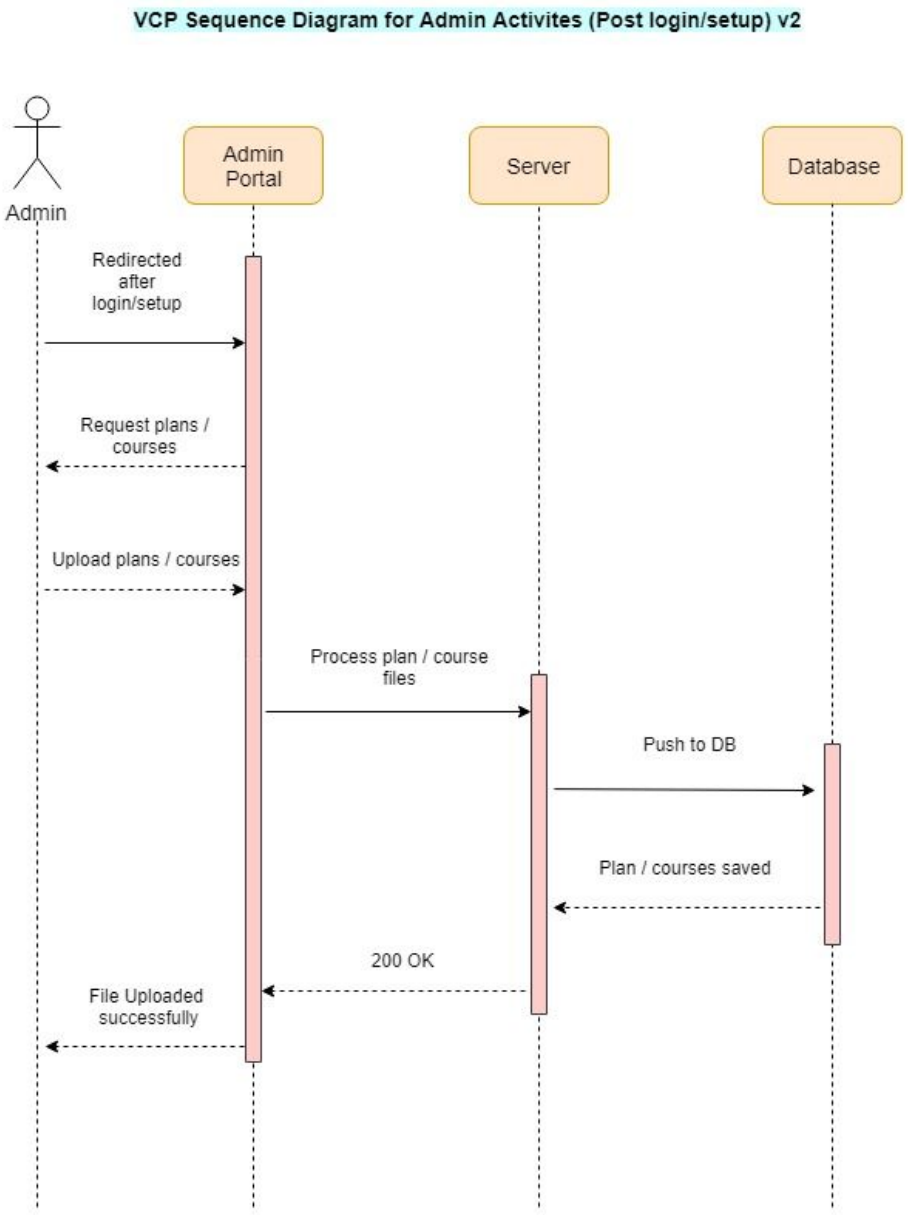
2.5.1 Use case: Sign up/Login for Admin/User



2.5.2 Use case: User Activities (Post Login)

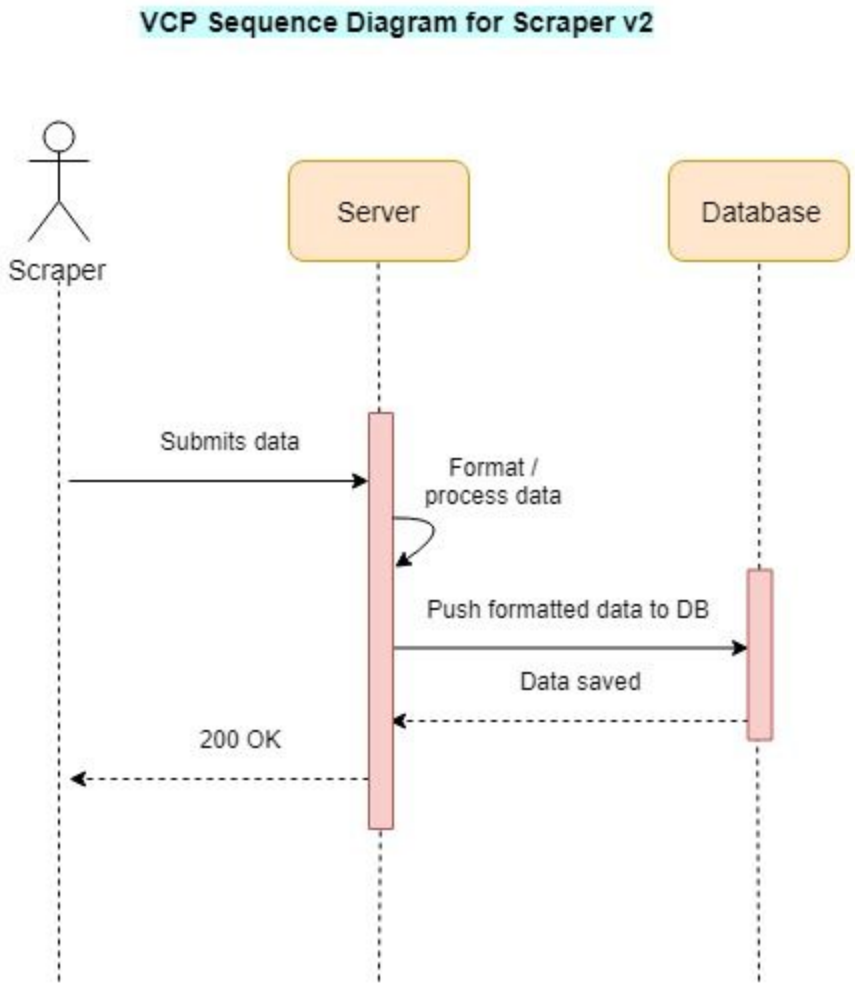


2.5.3 Use case: Admin Activities (Post Login)



Assumptions:
Admin uploads correctly formatted file

2.5.4 Use case: Scraper



3 USER INTERFACE

3.1 LIST OF FEATURES

1. Student has the ability to favourite degree plans
2. Student has the ability to save a degree plan with a desired name
3. Drag and drop functionality to move courses around semesters and year
4. Student can view and update multiple degree plans
5. Account creation : admin or student
6. Student can create and save a maximum of 10 degree plans
7. Student can declare previously taken courses upon account creation to tailor the VCP to the student
8. Degree and program selection

3.2 INTERFACE DESCRIPTION

The following is a brief description of the intended user interaction, the interface, and the functionality of the Visual Course Plan as it stands at this time.

3.2.1 LOGIN + SIGN UP PAGE

The Login Page is the first page everyone will interact with upon searching for the Visual Course Planner via a web browser. This page enables every user to either create or log into their account. The default menu will prompted the user to log into their already made account. A button will be provided to change the current menu to the account creation menu.

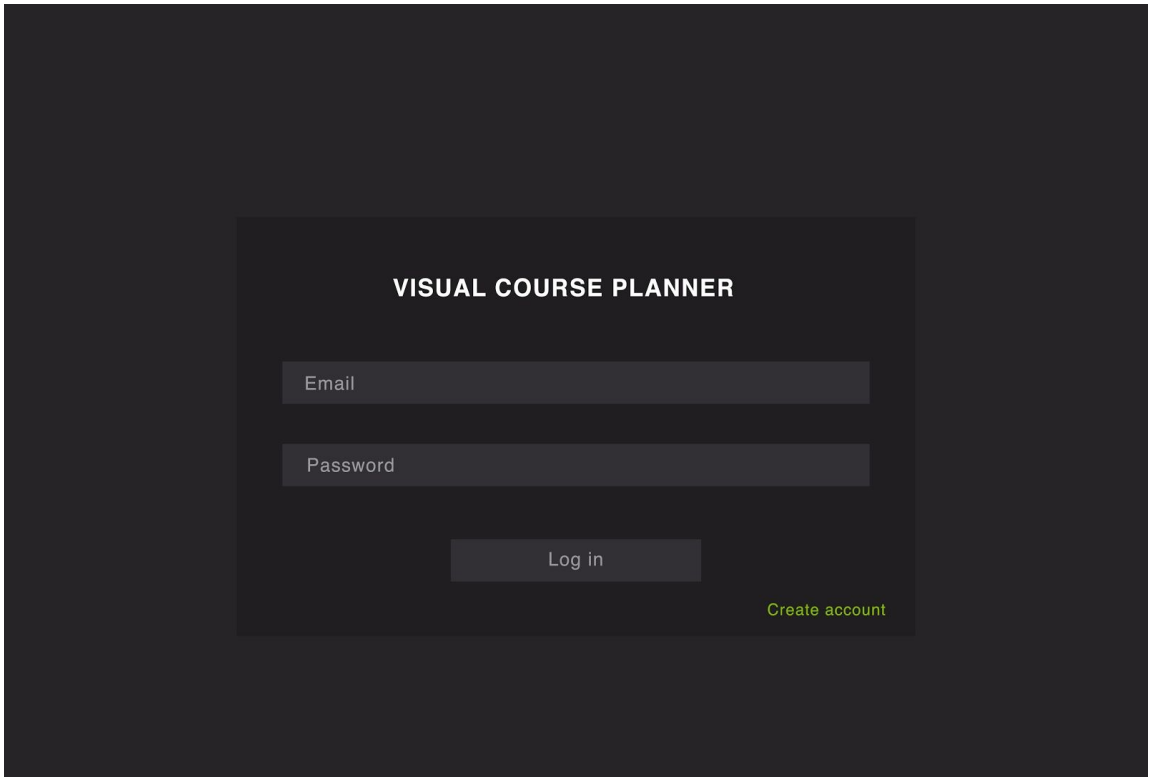
3.2.2 STUDENT INFORMATION PAGE

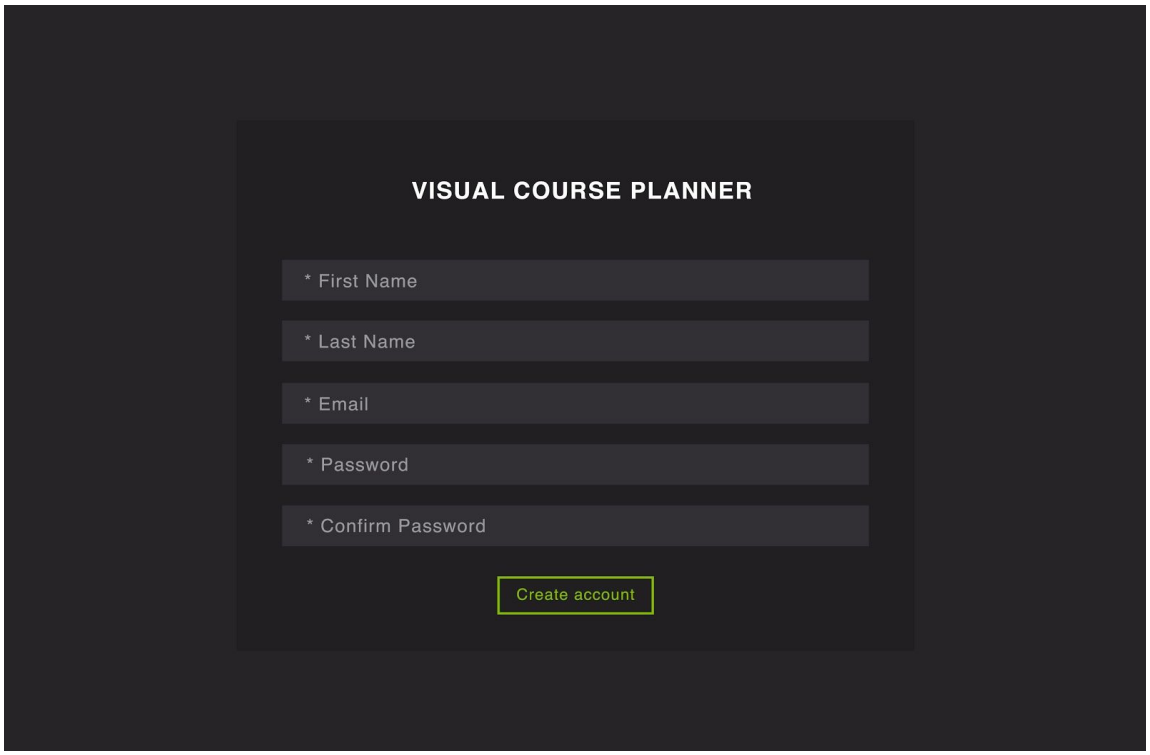
The Student Information Page is the next page a user will see if a new account it created. This page allows the student to input personal information to allow the Visual Course Planner to update specifically to the student. This personal information can include: previously taken courses (university and high school AP, IB etc), desired program, year the student is currently enrolled in. While selecting a degree, a description of the current selected degree is displayed - this description is provided by the University of British Columbia. A degree *must* be chosen in order to proceed to the VCP.

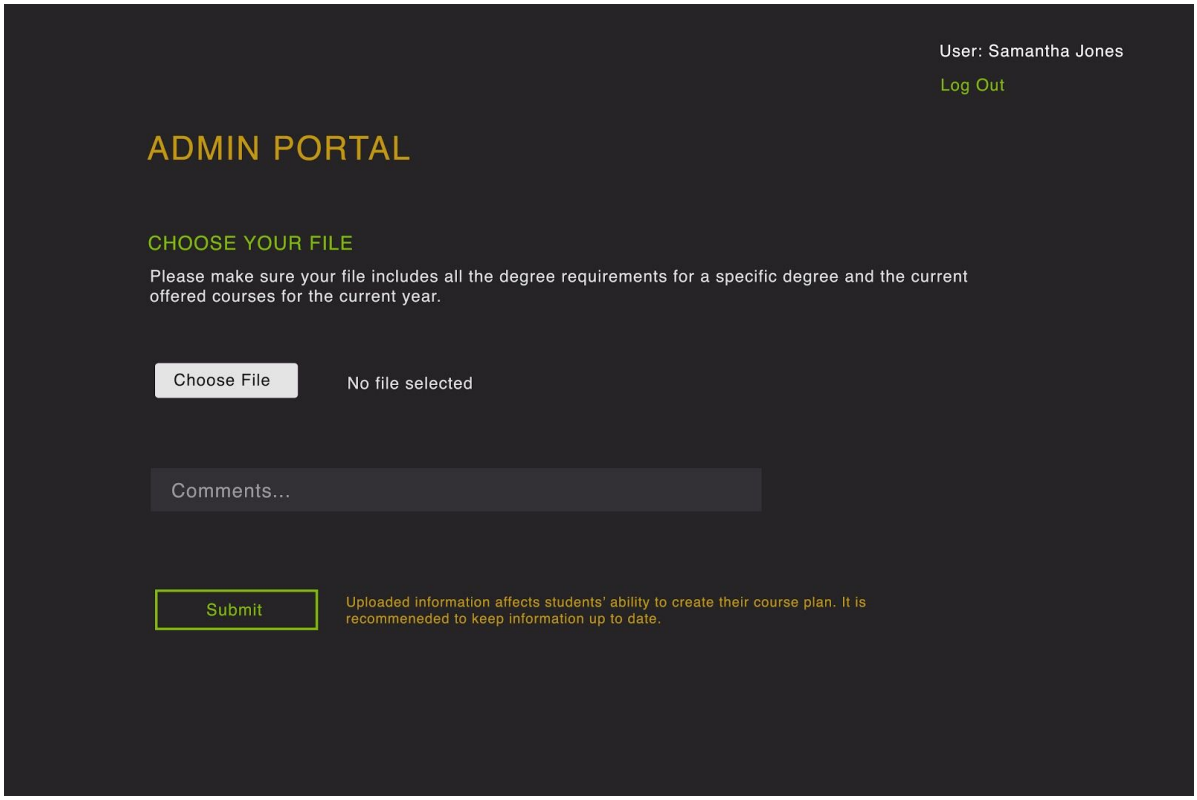
3.2.3 VISUAL COURSE PLANNER

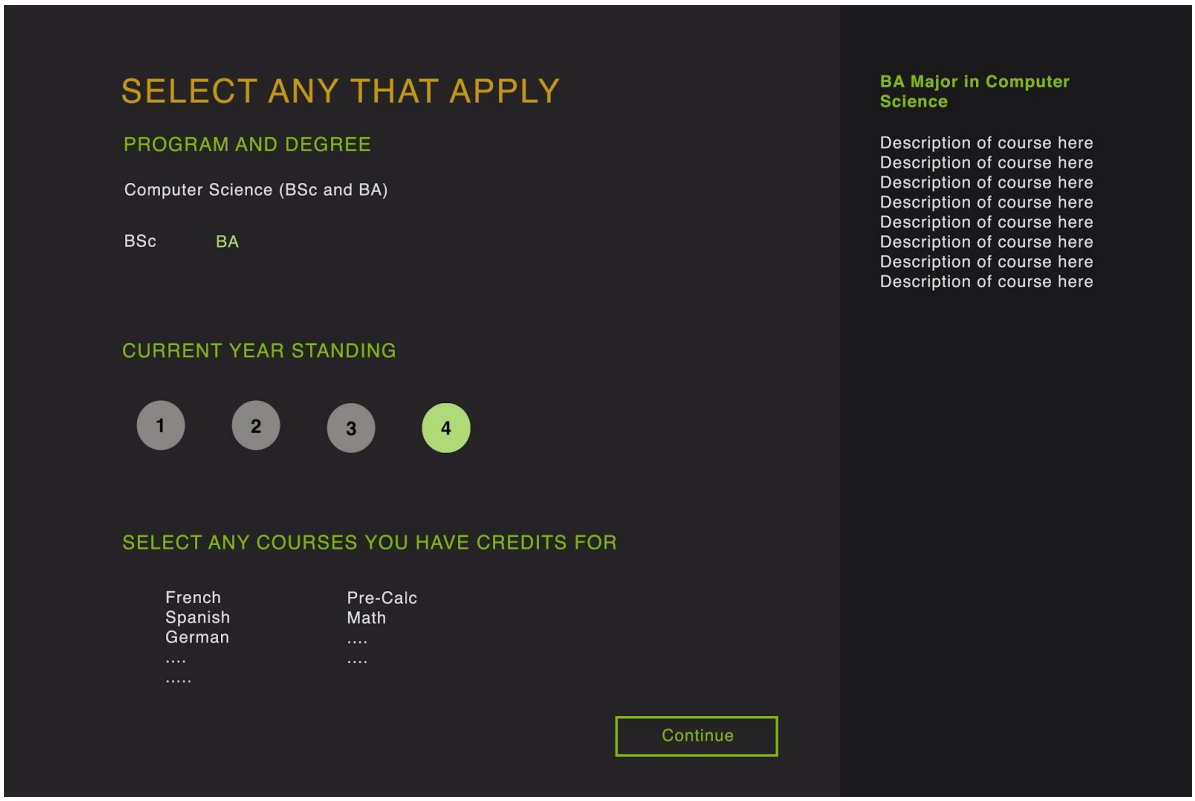
The Visual Course Planner is the main page of the entire program. This page allows students to fully personalize their entire degree plan during their time at UBC Okanagan. Here, a student can utilize the drag and drop feature of the VCP to move the required courses for their specified degree, around (via years, and semesters). Searching courses the student wishes to take (such as electives) and adding those courses to their degree is also an ability. Suitable warnings will be displayed if a recent change is incompatible with UBC Okanagan course calendar. These warnings could include a missing prerequisite but not limiting to. The student can also name their current active degree plan and favourite it for easy access upon a later session,

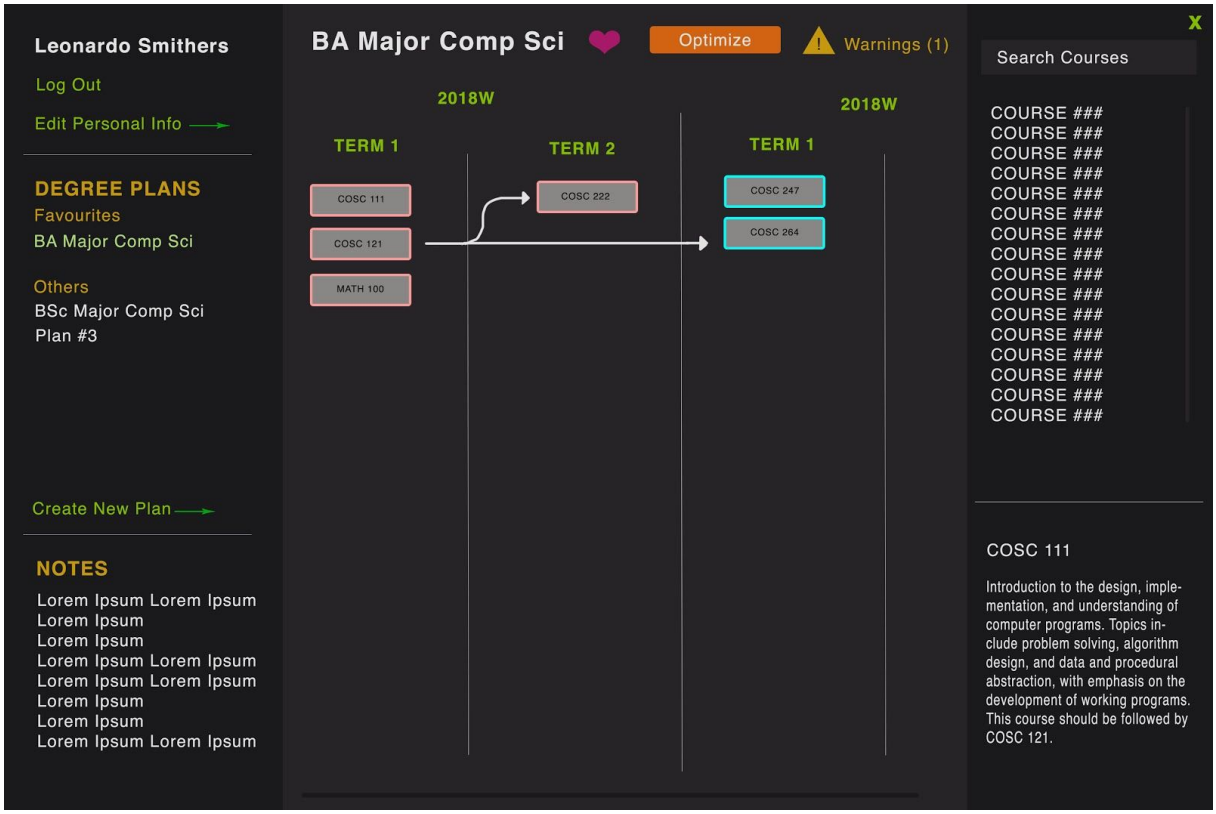
3.3 SITE MOCKUPS






User: Samantha Jones
Log Out
ADMIN PORTAL
CHOOSE YOUR FILE
Please make sure your file includes all the degree requirements for a specific degree and the current offered courses for the current year.
Choose File
No file selected
Comments...
Submit
Uploaded information affects students' ability to create their course plan. It is recommeneded to keep information up to date.


SELECT ANY THAT APPLY
PROGRAM AND DEGREE
Computer Science (BSc and BA)
BSc
BA
CURRENT YEAR STANDING
1
2
3
4
SELECT ANY COURSES YOU HAVE CREDITS FOR
French
Spanish
German
....
.....
Pre-Calc
Math
....
....
Continue
BA Major in Computer Science
Description of course here
Description of course here
Description of course here
Description of course here
Description of course here
Description of course here
Description of course here
Description of course here


Leonardo Smithers
Log Out
Edit Personal Info
DEGREE PLANS
Favourites
BA Major Comp Sci
Others
BSc Major Comp Sci
Plan #3
Create New Plan
NOTES
Lorem Ipsum Lorem Ipsum
Lorem Ipsum
Lorem Ipsum
Lorem Ipsum Lorem Ipsum
Lorem Ipsum Lorem Ipsum
Lorem Ipsum
Lorem Ipsum
Lorem Ipsum Lorem Ipsum
BA Major Comp Sci
Optimize
Warnings (1)
2018W
2018W
TERM 1
TERM 2
TERM 1
COSC 111
COSC 121
MATH 100
COSC 222
COSC 247
COSC 264
X
Search Courses
COURSE ###
COURSE ###
COURSE ###
COURSE ###
COURSE ###
COURSE ###
COURSE ###
COURSE ###
COURSE ###
COURSE ###
COURSE ###
COURSE ###
COURSE ###
COURSE ###
COURSE ###
COURSE ###
COSC 111
Introduction to the design, implementation, and understanding of computer programs. Topics include problem solving, algorithm design, and data and procedural abstraction, with emphasis on the development of working programs. This course should be followed by COSC 121.

4 LIMITATIONS

4.1 USER

1. The ability to print or export each degree plan created by a student will not be provided.
2. The ability to upload a profile picture or personalize the Students' account will not be provided

4.2 ADMIN

1. Admins do not have access to students' account or content
2. Admins cannot refer to previously uploaded files to the VCP as there is no kept record of uploaded files

4.3 ENVIRONMENT

1. The VCP is designed for a desktop browser only
2. The system will handle a maximum ~300 requests/hour.
3. No email warnings will be sent.
4. Personalizing any visual components of the Visual Course Planner will not be optional; set colours, fonts, shapes, animations are final.
5. Integration with another system (ie. UBC Okanagan website) will not implemented during the creation of the Visual Course Planner.
6. The connection of each students UBC account and visual course planner will not be implemented.

5 TEST/QA PLAN

The following is an outline of the format and standard of testing that will be done before, during, and after the development of the Visual Course Planner.

5.1 GOAL

The primary goal of the test plan is to ensure that the system meets the full requirements (including quality requirements), satisfies the use case scenarios, and maintains the quality of the product. The secondary objective is to minimize defects and errors during both development and the end product. The success of the project leans heavily on adhering to this document as written, but also being able to account for changes that occur during development and adapting to those changes by modifying the test document accordingly, in order to maintain quality. This initial draft of the test plan attempts to be as close as possible to a testing framework to start with.

5.2 SCOPE

The scope of testing will include all functions and external interfaces of the website.

5.3 OUT OF SCOPE

Performance and stress testing of the VCP will not be performed.

5.4 TECHNIQUES

Test-driven development will be adhered to. To do this, test for a component/piece of functionality will be written first in black-box testing style to ensure the system behaves as required. Then, code will be written to pass the test, one at a time. Refactoring will occur after the test has been passed.

We will be following a testing pyramid. The majority of our tests being unit tests, with some integration tests and few system level tests. This will ensure both thorough testing and fast speed of test running, making the feedback loop of development workflow to test running as fast as possible.

5.4.1 Unit Testing

Unit Testing will ensure the lowest level of functionality within the app (ie a unit) is working as expected. For instance, we could test that a getCourse function returns the correct formatting of a course item. Unit testing will be done before each component is built.

5.4.2 Integration Testing

While unit tests will ensure each component works in isolation, integration tests will ensure components work together. For instance, we can user account creation, given valid account information.

5.4.3 System Testing

System testing will be done to ensure the behaviour of the entire system is done according to the specification. We will do manual testing, such as creating a course plan ourselves, and use Selenium to automate the process.

5.4.4 User Testing

User testing will allow us to get feedback from users once the MVP is complete and improve the experience based on that feedback. In a user testing session, we can ask a user to accomplish a task - such as “Create a degree plan with 2 courses” - without prior knowledge of the application, and see how easily they accomplish the task. If they fail to accomplish the task, or take longer than a predetermined time, we know the experience needs to be streamlined or improved.

5.5 FEATURE CRITERIA

All features listed here refer to the feature breakdown in the Project Charter. For a feature to be considered done, it must pass all associated tests.

Feature 3.4: User authentication				
ID	Title	Pre-conditions	Test Steps	Expected Results
T001	User logs in	User is on the website User is logged out User has an account	1. Click login button 2. Enter a correct username 3. Enter a correct password 4. Click log in	The main page of the VCP is displayed to the user. The user is logged in, indicated by text showing their name.
T002	User creates	The user does not exist in	1. Click signup	An email is sent to

	account	the system	2. Enter a unique email, first name, last name, and password 3. Click sign up	the user's email address. The email contains a valid link to confirm account creation.
T003	User logs out	User is logged in	1. Click logout button	The user is logged out, and taken to the login page
T004	Incorrect log in	User is not logged in	1. Enter an invalid email or password	The user is not logged in. A message is displayed to the user indication which field is incorrect.
T005	Secure password storage	User has clicked create account User has entered valid account information	1. Send the password to the database	The password in the database is salted and hashed
T006	Confirm account	User has a confirmation email in their inbox	1. Click the link in the email	The user's account is confirmed. The user is taken to an interface asking for more information. A message is displayed that says their account is now valid.
Feature 3.7: Admin file upload				
ID	Title	Pre-conditions	Test Steps	Expected Results
T007	Admin can upload file	User is logged in User is an admin User is on Admin Portal interface	1. Click upload file button 2. Select file 3. Click submit button	File is uploaded. Displays message: "file uploaded successfully"
T008	Admin uploads incorrect format	Admin is logged in Admin has just uploaded a file The file is in the wrong format	1. Click upload file button 2. Select file 3. Click submit button	Displays error message: "Incorrect file format"
T009	File storage	Admin has clicked submit button File is in correct format	1. Send file to server	File is stored on the server. The file is in a folder called "UBCO_Current_Courses"
T010	File data import	Admin has uploaded file to server	1. Process the uploaded file	Data from the file is correctly parsed. The data is stored in the database

Feature 3.8: Auto save plan				
ID	Title	Pre-conditions	Test Steps	Expected Results
T011	Plan modified	User is logged in User is editing a plan	1. Make a change to the plan 2. Don't make any changes for 5 seconds	The plan is sent to and stored in the database
Feature 3.9: Prereq missing warning system				
ID	Title	Pre-conditions	Test Steps	Expected Results
T012	User adds a course with missing prereqs	User is logged in User is editing a plan The prerequisites for the course do not exist in the plan	1. Add a course to the plan	The warning summary displays a message that the added course is missing prerequisites.
T013	User adds a course with correct prereqs existing	User is logged in User is editing a plan Course prereqs exist	1. Add a course to the plan	The warning summary does not change. Course is successfully added to plan
T014	User adds a course into its corresponding term	User is logged in User is editing a plan Course prereqs exist	2. Add a course to the plan	Course is added to plan in its corresponding term
T015	User removes a course with missing prereqs	User is logged in User is editing a plan The prerequisites for the course do not exist in the plan Warnings are displayed about the missing prereqs	1. Remove a course from the plan	The warning indicating "missing prereqs" are removed.
Feature 4.1: Admin file upload interface				
ID	Title	Pre-conditions	Test Steps	Expected Results
T016	Admin Portal interface is displayed	User is logged in as admin	1. Login as admin	The Admin Portal interface is displayed to the user.
Feature 4.2: Course list display				
ID	Title	Pre-conditions	Test Steps	Expected Results
T017	Course list is displayed	User is logged in	1. Navigate to the VCP interface	The courses are displayed in a sidebar on the right of the screen.

Feature: 4.3: Course list filter				
ID	Title	Pre-conditions	Test Steps	Expected Results
T018	Course list filters courses	User is logged in	1. User types text into the search bar	The course list is filtered with only courses that match the entered text.
Feature 4.4: Login/signup screen				
ID	Title	Pre-conditions	Test Steps	Expected Results
T019	Login screen display	User is on the VCP website User is not logged in	1. Click login	The login screen is displayed
T020	Signup screen displayed	User is on the login page User is not logged in	1. Click create account	The create account screen is displayed
Feature 4.5: Degree history input				
ID	Title	Pre-conditions	Test Steps	Expected Results
T021	User enters more information	User has just created an account	1. Enter degree information, major information, previously taken courses (using the course filter) 2. Click "Continue" button	The users data is sent to the database. The page displays the VCP page.
Feature 4.7: Semester view				
ID	Title	Pre-conditions	Test Steps	Expected Results
T022	Semester view display	User is logged in User has a pre-existing course plan	1. Navigate to the VCP interface	The courses are displayed in columns for each semester.
T023	Semester is added	User is logged in User is on VCP interface	1. Click the "Add semester" next to the last semester added	The semester is displayed to the right of the last semester added
T024	Semester is removed	User is logged in User is on VCP interface	1. Click the minus symbol next to the semester name	The semester column is removed from the display
Feature 4.8: Course prerequisite rearrange				
ID	Title	Pre-conditions	Test Steps	Expected Results
T025	Plan is rearranged on addition of course	User is logged in User has more than one course in the plan	1. Add course with prerequisites that are already in the plan	Course layout is arranged so that the prerequisite arrows are not

				overlapping.
T026	Plan is rearranged on removal of course	User is logged in User has more than one course in the plan	1. Remove course with prerequisites that are already in the plan	Course layout is arranged so that the prerequisite arrows are not overlapping.
Feature 4.9: Course display				
ID	Title	Pre-conditions	Test Steps	Expected Results
T027	Courses are displayed	User is logged in User has a degree plan The degree plan has courses stored in it	1. Select the degree plan	The degree plan should display all courses in it
T028	Course is added	User is logged in User is editing a degree plan	1. Drag course into plan	The course should be added to plan
Feature 4.10: Plan favouriting				
ID	Title	Pre-conditions	Test Steps	Expected Results
T029	Favourite a plan	User is logged in User is editing a degree plan	1. Click the “Favourite plan” button next to degree plan name	The plan is displayed in the favourites section in the list of plans. The “Favourite plan” button changes to say “Remove from favourites”. The change are saved in the database
T030	Unfavourite a plan	User is logged in User is editing a degree plan Plan is a favourite	1. Click the “Remove from favourites” button	The plan is removed from the favourites section in the list of the user’s plans. The “Remove from favourites” button changes to say “Favourite plan”
Feature 4.11: Plan notes				
ID	Title	Pre-conditions	Test Steps	Expected Results
T031	Edit plan note	User is logged in User is editing a degree plan	1. Edit text in “Notes” section	The notes for the plan are automatically updated in the database
Feature: 5.2 Course importer from scraper				
ID	Title	Pre-conditions	Test Steps	Expected Results

T032	Courses are imported	Scraper has retrieved course data	1. Parse data 2. Build models 3. Send models to database	The courses are stored in the database with correct course and session information
Feature 8.4: Optimization button				
ID	Title	Pre-conditions	Test Steps	Expected Results
T033	Optimize course plan	User has a course plan with at least one course There are no warnings on the plan	1. Click "Optimize" button	The courses are compressed into the shortest time frame

5.6 QUALITY PLAN

To ensure product quality we are enforcing a git branching workflow and code merging pattern.

5.6.1 Branching Strategy

During this project are following a master -> dev -> feature branch based workflow.

5.6.1.1 Master

The master branch ("master") will contain production ready code, that has been fully tested according to the quality plan. This is where code will be deployed from to the remote server. Pull requests to master need to be approved by the Integration Lead before being merged. Only pull requests from dev will be merged into master, except in the case of hotfixes. PRs will need to pass CI builds in Travis.

5.6.1.2 Dev

The dev branch ("dev") will be where features are merged together and tested in CI. Merges to dev will need to be approved by at least one other team member, and will need to build in CI before being merged..

5.6.1.3 Feature branches

Feature branches will be prefixed with "feature/". Each branch will contain independent feature of the system. All pushes to a feature branch will automatically trigger a build of the branch in CI.

5.7 TOOLS

Mocha will be used for unit and integration testing of JavaScript code. Selenium will be used for automating browser tasks and testing.

6 TECHNICAL SPECIFICATION

The VCP will be built using MySQL, Express.js, React, and Node.js. The codebase will be stored remotely on GitHub. Continuous Integration will be provided by Travis CI, and the system will be deployed to Heroku.