**C868 – Software Capstone Project Summary**

**Task 2 – Section A**



|  |  |
| --- | --- |
| **Capstone Proposal Project Name:** | CCC SAYOP Registrar – Mobile Application |
| **Student Name:** | Greg Westmoreland |

# Table of Contents

[Table of Contents 2](#_Toc92714319)

[Business Problem 3](#_Toc92714320)

[*The Customer* 3](#_Toc92714321)

[*Business Case* 3](#_Toc92714322)

[*Fulfillment* 3](#_Toc92714323)

[Existing Gaps 4](#_Toc92714324)

[SDLC Methodology 4](#_Toc92714325)

[Deliverables 5](#_Toc92714326)

[*Project Deliverables* 6](#_Toc92714327)

[*Product Deliverables* 6](#_Toc92714328)

[Implementation 7](#_Toc92714329)

[Validation and Verification 7](#_Toc92714330)

[Environments and Costs 8](#_Toc92714331)

[*Programming Environment* 8](#_Toc92714332)

[*Environment Costs* 8](#_Toc92714333)

[*Human Resource Requirements* 9](#_Toc92714334)

[Project Timeline 9](#_Toc92714335)

# **Business Problem**

***The Customer***

The customer, Cornerstone Community College (CCC), is a public community college with 3 campuses in Cornerstone County, Texas. It serves more than 3,000 students annually in degree-granting, continuing education, and adult education programs. CCC offers associate degrees and career/technical certificates in more than 50 areas of study. CCC’s mission is to provide affordable and open access to quality learning that will lead to further community enrichment.

## 

## ***Business Case***

CCC offers a special program called “Study at Your Own Pace” (SAYOP) which allows current students the flexibility of dividing up the Spring and Fall semesters into one-month terms that last an average of 30 days. This program provides students with an opportunity to comprehensively break down the area of study into smaller sizes. This, in turn, allows them the opportunity to complete their degree and/or certificate quicker than the normal two semesters per year template.

CCC allows current students to register for **SAYOP** classes through a mobile application platform called **CCC Scheduler App version 1.0**. Using this mobile app, students are allowed to create a term name, start date, and end date of their choosing. Once the term has been created the student has full flexibility to add/drop/edit courses associated with that term, along with assessments associated with that course.

CCC has been using version 1.0 for over a year, and they have been receiving complaints from the students participating in the SAYOP program. Most of the complaints have referenced difficulties with the user interface, as many users find it difficult to navigate within the app.

## ***Fulfillment***

A new and improved mobile app will be created called **CCC SAYOP Registrar version 2.0**. This system will interface with ROOM persistent library as an abstraction layer to the SQLite database that will run on an Android tablet or phone. Students will be able to download the mobile app from the Google Play Store and will be able to begin using it immediately upon downloading. The preexisting framework requirements will be handled by the Google Play Store.

The basic functionality existing in the previous version will remain the same. Students will be able to create, edit, or delete terms. They will be able to create, edit, or delete a course. They will be able to create, edit, or delete an assessment. They will be able to set notifications for start and end dates for a term, course, or assessment. They will be able to share information about a term, course, or assessment via Email or SMS text message.

# **Existing Gaps**

The new mobile application, CCC SAYOP Registrar, will be adding four specific improvements and features that differ from the original version currently in use.

* Search functionality
* Capability of generating various reports:
  + Display all terms
  + Display all courses
  + Display all assessments
  + Display instructor information
* Exception controls and validation functionality for handling of user error
* User login and password interface
* Improved menu navigation

These improvements will address the complaints reported by users of version 1. As an example, the students will be able to navigate to different screens with the improved menu being introduced in version 2.

# **SDLC Methodology**

The SDLC Methodology utilized for this project will be the Agile method. Using this approach will be beneficial because it will allow a collaborative effort with cross-functional departments working together for the completion of the project. Through the iterative release process and definable milestones, the agile method is the quickest, most cost-effective and logical approach for delivery to CCC.

The following phases that will occur during the Agile method:

* Requirements Phase
* Analysis Phase
* Design Phase
* Coding Phase
* Testing Phase
* Maintenance Phase

During the **requirements** phase, the Project Manager (PM) will gather all the specifications from CCC. This phase is designed to incorporate the set of new features that will be added to the mobile application in version 2.0. The requirements gathered will be organized, labeled, prioritized, and distributed to all project stakeholders by the PM.

During the **analysis** phase, the PM will break down the requirements into smaller, simpler tasks called stories. Each story will be assigned a time estimate of story points (1 story point is equivalent to 1 day). The stories are then organized and distributed to teams for completion during sprints (one sprint is equivalent to 2 weeks). Each team will complete several stories based on their available bandwidth and team size. Milestones and target due dates are assigned for each feature that will be added for the project.

During the **designing** phase, the technical specifications will be discussed, such as which programming language, database diagram, deployment release schedule, and which testing strategies (unit testing, integration testing, system testing, acceptance testing, performance testing, security testing, usability testing, and compatibility testing) will be used.

The deliverables completed in this phase will allow the project manager to sketch how the user interface will look and respond with a low-fidelity wireframe, to be passed onto the software developers that will provide the development code for the project. Later in the project, a high-fidelity prototype will demonstrate the User Interface (UI) design and demonstrate the relationship between the data models and testing plans to ensure a fully functional application.

The **coding** phase is when the actual development of the mobile application will take place, with software developers writing and testing code. The code will be stored in the company’s git hub repository for continual integration, continual delivery (CICD) into their staging environment. This phase will also use automated unit testing through the development process.

The **testing** phase is when quality engineers will test each story in its own development branch through black-box testing, white-box testing, automated testing, and ad-hoc testing. Subsequently, retesting will be done again with all integrated stories that are part of one release for each sprint in the release testing branch. Any stories that do not pass testing will be sent back to the developer for rework. Any new bugs will either be fixed in the existing branch or opened as a new story with a dependency upon the original story. With everything passing for a release, the integrated stories are merged into a staging branch for continual automated testing.

During the **maintenance** phase the PM will communicate with CCC when new milestones/deliverables are released to the mobile app. CCC will communicate with the PM with any undesirable results that will need to be addressed. Bugs and/or corrections in this phase can be very costly and require additional work to be done by the PM, developers, and QE personnel.

At the end of each Sprint period, a retrospective will be given on what was done well and what could be done better for the next Sprint. A new sprint planning meeting is held to ascertain what stories remain to be done from the backlog.

# **Deliverables**

There are 6 types of deliverables that are associated with the Agile SDLC method that CCC has requested for the CCC SAYOP Registrar Mobile Application:

## 

## ***Project Deliverables***

* Requirement’s Document:
  + This outlines the features and functionality of the mobile app in detail and represents the software developer’s understanding of the stakeholders needs and expectations.
  + This needs to be written clearly and specifically as to not add any project creep for what is expected upon final delivery of the product.
* Project Schedule
  + This lists a detailed timeline for each deliverable milestone for the project.
  + Each milestone is followed by a description, expected functionality, acceptance testing, and due date.
* Low-fidelity Wireframe
  + A mockup of the entire application is designed to demonstrate the workflow of the mobile application.
  + The mockup does not need to be colorized or functional, but it should demonstrate the navigational flow of the mobile application.
* High-fidelity Prototype
  + Originating from the mockup, this prototype will focus on the look and feel of the mobile app, by displaying color scheme, functional elements within the prototype that demonstrates basic usage and navigational flow.
* Database Design
  + A model will demonstrate the database schema, including structures, data types, and relationships between the different tables used for the application.
* Test Plans
  + A testing document will lay out all the unit tests to validate functionality of the application in its entirety. These unit tests will test the newly designed features of the application through white-box and black-box testing methods. The unit tests will be done through Jenkins automated building of the development branch, building of the release branch, and be done prior to being released into the staging branch.
  + Specific functional and integration testing will be done by Quality Engineers through a variety of testing methods, that include automated scripts and ad-hoc testing with predefined, measurable outcomes.

## ***Product Deliverables***

* Functional database that matches the table design laid out in the database schema to meet the mobile application’s need.
* A fully developed mobile application that meets all functionality specified by CCC stated in the requirements document.
* Wireframe, a low-fidelity, rough representation of the mobile application showing navigational flow.
* Prototype, a high-fidelity UI with partial functionality that will demonstrate the requested features laid out in the requirements document.
* Deployment to a secure, staging environment available to all stakeholders stored on the cloud.
* Full support with training and maintenance for CCC staff members during each stage of the project.

# **Implementation**

The implementation of this project for each iteration will be divided into three stages. During the first stage the development team will create a developmental .apk file from Google that will allow the developer to build the mobile app directly on an Android device (phone/tablet) for testing after each iteration release.

The second stage allows the development team to upload a beta version of the mobile app to the Google Play Store allowing stakeholders to provide acceptance testing after each milestone is reached and delivered. The beta program on the Google Play Store will not allow the general public to access the beta version of the mobile app.

The third stage will consist of transitioning from a beta release version to a production release version made available for downloading and installation from the Google Play Store. This will allow the general public to begin using the new mobile application. At this point, the final maintenance phase begins along with final training and support being provided to IT staff and registrar’s office staff at CCC.

# **Validation and Verification**

A comprehensive testing plan will be outlined in a testing document to ensure that all requirements stated in the Requirements Document are properly tested and functioning as expected. Testing and retesting are an integral part of the development and delivery of the new CCC SAYOP Registrar Mobile App.

During the coding phase, each developer will create unit tests before any new methods are written for the mobile application. Any bugs, defects, or improvements will be tracked in the Atlassian project software called JIRA®. The project manager will work these tickets / stories into 2-week sprint periods.

After completion of a story, the testing team of quality engineers will teach each individual development branch on an android emulator, testing various operating system levels. They will test individual functionality specified with the story of the issue they are testing. They will also regression test through automated testing scripts and random ad-hoc testing.

Once an issue has passed QA testing, the story will be integrated into other stories within a release branch and the original test plan will be repeated. Upon passing, the integrated branch will be released into a staging branch where stakeholders are able to test out functionality after each iteration cycle.

# **Environments and Costs**

## 

## ***Programming Environment***

The application will be developed in Android Studio or IntelliJ using the Java language as the base language. The database will utilize the ROOM persistent library as an abstraction layer to the SQLite db that will be stored on the Android device.

The application will be stored in a company GitHub repository and individual branches can be cloned on a secure development application server or on an individual developer’s computer, where changes can be made before committing it to their JIRA branch.

Software Used for The Project:

* JIRA
* GitHub
* Confluence
* Android Studio / IntelliJ
* Slack Instant Messenger
* Microsoft Exchange for Email
* Cisco WebEx For Collaboration / Meetings

Hardware Used for The Project:

* Each team member will have their own laptop (Dell or MacBook Pro)
* Each team member will have an Android Tablet – Samsung Galaxy A7 Lite
* Each team member will have an Android Phone – Samsung Galaxy S Series

## ***Environment Costs***

The associated cost of the project to be hosted on the Google Play store is covered with no additional fees as our company pays a yearly membership fee to Google to store our development and production products there.

The advertising plan that was agreed upon will be hosted and conducted through a third-party vendor with the estimated cost of $8,500.00 per month using Google, Facebook, LinkedIn, and Twitter social media advertising.

## ***Human Resource Requirements***

* 1 x Project Manager
* 1 x UX Engineer
* 1 x Development Manager
* 1 x UI Android Software Engineer
* 4 x Software Developers
* 3 x Quality Engineers

|  |  |  |
| --- | --- | --- |
| **Resource** | **Rate x Time** | **Total** |
| Project Manager | $62.00 x 70 hours | $ 4,340.00 |
| UX Engineer | $88.00 x 12 hours | $ 1,056.00 |
| Development Manager | $92.00 x 70 hours | $ 6,440.00 |
| UI Android Software Engineer | $68.00 x 20 hours | $ 1,360.00 |
| Software Developer | $76.00 x 120 hours | $ 9,120.00 |
| Quality Engineer | $52.00 x 75 hours | $ 3,900.00 |
| **TOTAL Human Resource Cost** |  | **$ 26,216.00** |

# **Project Timeline**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Phase | Milestone/Task | Deliverable | Description | Dates |
| Requirements | Task 1 | Draft Requirements Document | Meeting with stakeholders and approve requirements document | 1/1/2022 – 1/4/2022 |
| Analysis | Task 2 / Breakdown | User Stories, Due Dates, Milestones | Project manager will break down requirements into the simplest tasks (stories), assign story points, prioritize, divide stories up among teams, schedule delivery dates, write out milestones and dates | 1/5/2022 – 1/25/2022 |
| Designing | Task 3 / Design | Low-fidelity wireframe    High-fidelity prototype | UX designer will create a low-fid wireframe to show navigation.  A high-fid prototype will be developed by UI Android Engineer showing new app features | 1/26/2022 – 2/14/2022 |
| Coding | Task 4 / Feature 1 | Exception Controls and Validation Functionality | The development team will write in exception handling and validation input functionality to handle error input from the user based off the requirements document. | 2/14/2022 – 2/24/2022 |
| Testing | Task 5 / Test Feature 1 | QE Testing | QE Testing will test all exception handling and validation functionality introduced to make sure proper error messages and exceptions messages are clear and precise | 2/24/2022 – 2/28/2022 |
| Maintenance | Task 6 / Milestone | PM communicates to stake holders | The stake holders are shown the new features, acceptance testing is done in the staging environment after release | 03/01/2022 – 03/14/2022 |
| Coding | Task 7 / Feature 2 | Easy Menu Navigation | The development team will redesign and implement a new menu navigation system for the mobile app | 03/01/2022 – 03/10/2022 |
| Testing | Task 8 / Test Feature 2 | QE Testing | QE Testing will test all of the new navigational menu throughout the entire app | 03/10/2022 – 03/14/2022 |
| Maintenance | Task 9 / Milestone | PM communicates to stake holders | The stake holders are shown the new features, acceptance testing is done in the staging environment after release | 03/15/2022 – 03/31/2022 |
| Coding | Task 10 / Feature 3 | Generate Reports | The development team will introduce 4 new reports that will allow users to display all terms, all courses, all assessments, and all instructors. | 03/15/2022 – 03/28/2022 |
| Testing | Task 11 / Test Feature 3 | QE Testing | QE will test the 4 new reports to verify that they are functioning according to the requirements document. | 03/28/2022 – 03/31/2022 |
| Maintenance | Task 12 / Milestone | PM communicates to stake holders | The stake holders are shown the new features, acceptance testing is done in the staging environment after release | 04/01/2022 – 04/15/2022 |
| Coding | Task 13 / Feature 4 | Search functionality | The development team will implement a search functionality to allow searching by term name, course name, assessment name, start dates, and end dates. | 04/01/2022 – 04/10/2022 |
| Testing | Task 14 / Test Feature 4 | QE Testing | QE Team will test the new search functionality by term name, course name, assessment name, start date, and end date as described in the requirements documents. | 04/10/2022 – 04/15/2022 |
| Final Maintenance | Task 15 / Milestone | PM communicates to stake holders | The stake holders are shown the new features, acceptance testing is done in the staging environment after release. Beta release is done for select IT staff / Registration staff of CCC. Training provided for all CCC staff. Production Release scheduled for 06/01/2021 | 04/15/2022 – 05/31/2022 |