C868 – Software Capstone Project Summary

Task 2 – Section A

Capstone Proposal Project Name: CDUTermTracker- Term Tracking Application for CharDennis University

Student Name: Shannon Marie Peck

Table of Contents

Contents

Business Problem

The Customer

The customer is CharDennis University, an online university specializing in technical degrees. It has a student body of 10,000 students on average, and predicts growth of 5% this year. The university currently utilizes web-based term tracking for its students, and has no mobile infrastructure. CDU's mission is to provide a high-quality, student-driven education that is self-paced and comprehensive, and to empower students to take charge of their schedule and learn at a pace which is most appropriate for them.

CDU's short-term goal is to provide a scalable mobile application that allows students to track their term progress. Their long-term goal is to leverage this mobile application into higher perceived quality by students, leading to future referrals and an increase of 10% in 2023.

Business Case

CDU's student body currently has no way to track their term progress on mobile devices, as the website used by CDU is not responsive and therefore scales poorly to mobile devices. An increasing number of students have expressed interest in access to term tracking on their mobile devices. The CDUTermTracker application will meet the client's needs by providing a high-quality application that will allow students to track their term progress.

Due to regulations, CDU's students are permitted to take no more than 6 courses per term. Each course has two assessments: a Performance Assessment and an Objective Assessment. The application will need to allow students to track their terms accordingly.

Fulfillment

CDUTermTracker is a multi-screen Android mobile application that will allow students to enter:

- Unlimited Terms
    - A term start date
    - A term end date
- Up to 6 courses per term
    - A course start date with optional notifications
    - A course end date with optional notifications
    - A course due date
    - Instructor information:
        - Instructor name
        - Instructor email
        - Instructor phone number
    - Optional, shareable notes
    - The ability to search courses by course name
- One Performance Assessment per course
    - An assessment due date with optional notifications
- One Objective Assessment per course
    - An assessment scheduled date with optional notifications
    - A field for Pre-assessment score

- The ability to generate reports based on scheduled date of upcoming assessments

Notifications should trigger if the associated date is within one week, and all fields that allow text should have security protections to defend against SQL injection attacks. All fields except notes are required, and should have error handling to prevent poorly formatted entries.

The application will use a SQLite database to store information locally on the user's mobile device, and will allow students to create, view, update, and delete terms and courses, and view and edit assessments. There will be additional error-handling to ensure that start dates do not occur after due dates.

Existing Gaps

CDU does not currently have a mobile application in place. Their current term tracking utilizes a website, which is not responsive to mobile screens. CDUTermTracker will provide CDU's students with a mobile application for term tracking.

SDLC Methodology and Deliverables

Agile methodology will be utilized when delivering this project due to this methodology's focus on delivering high-quality deliverables quickly and repeatedly iterating to improve those deliverables. Testing will be paramount and will take place repeatedly throughout the progress.

One benefit of Agile SDLC methodology is that it produces a working deliverable more quickly. This gives the customer an earlier opportunity to provide input to that deliverable, allowing our team to be more responsive to the customer's needs.

Below is an outline of the phases of Agile software development, the activities that take place in each, a high-level description of deliverables for each phase, and specific deliverables associated with this project.

Phases of Agile Methodology

Requirements Gathering

During this phase, the customer's needs are discussed and documented. Once the first iteration is complete and the product has been reviewed, this phase will be re-entered based on the results of the review.

Deliverables: On the first iteration, the deliverable is the documentation of client's requirements. On later iterations, documentation from the Review phase and any additional requirements from the user.

Project Deliverables: Requirements document

Design

In this phase, the client's needs have been determined, and the requirements documented. The program can now be designed, starting with a wireframe to plan layout and UML to design the program's structure. A test plan is also designed using the UML class diagram.

Deliverables: Wireframe, UML, and test plan in the first iteration. The deliverables for future iterations will be dependent on the previous stages' findings.

Project Deliverables: Low-fidelity wireframe and UML, and a test plan including: functional, integration, and unit tests

Development

Based on the layout and UML created in the Design phase, development can begin, and classes and functionality of the program implemented.

Deliverables: For the first iteration of this project, a fully functional application without fine-tuned design will be delivered. The second iteration's development phase will deliver a product with a polished and user-friendly design. During this phase, documentation including a user guide is created or updated.

Project Deliverables: Functional prototype based on the UML in the first iteration, in the second iteration a functional prototype with design implemented based on the wireframe. A user guide, updated if necessary with each iteration.

Testing

As functionality is implemented, testing occurs. If any test(s) reveal a problem, the development phase can be reentered to address that failure.

Deliverables: Results from testing.

Project Deliverables: Functional, integration, and unit testing summaries

Deployment

The initial deployment will be to a small group of internal Quality Assurance testers for testing of functionality, and subsequent iterations' deployment phases will be to increasingly large subsets of all users. 25 in the second iteration, 500 in the third, 1,000 in the fourth, and finally to the general student body.

Deliverables: The results of UAT testing, including bug reports and user input.

Project Deliverables: Bug reports and user input reports.

Review

The results of the user acceptance testing are reviewed and based on those results the requirements gathering phase is entered once again. Then the project enters the maintenance phase (which also follows the Requirements Gathering, Design, Development, Testing, Deployment, and Review phase structure).

Deliverables: The input from the previous phases and action items to address in the next iteration.

Project Deliverables: Documentation from previous phases that will guide the next iteration.

Implementation

As this application is not replacing an existing system and the client's priority is user experience, implementation will occur in a series of iterations as set forth in the Agile Software Development Life Cycle. This will allow us to rapidly respond to tester and user feedback with minimal frustration that frequent updates can cause.

Initial implementation will occur after the Testing phase of the lifecycle, during which validation and verification take place by internal QA tester. This first iteration's deployment implementation will involve a small group of internal testers who will focus on functionality.

The SDLC then starts a second iteration, including additional validation and verification in the Testing phase, based on the findings of that first iteration. This iteration's implementation will be a beta release to a small subset of 25 students who will begin initial End User Acceptance testing and offer input into design and functionality.

Their UAT results and reviews are used as input to the third iteration, which culminates in a larger release to 500 students for end user acceptance testing.

Based on their input, the fourth iteration is entered, and its deployment implementation will be to 1,000 students for additional UAT testing as well as load testing with a larger subset of users. The fifth iteration will implement deployment to all users, and additional iterations will be in response to user reported bugs or routine software maintenance.

Throughout each iteration of the SDLC lifecycle, all UAT test results and user input will be made available to stakeholders to facilitate stakeholder engagement and ensure their needs are being surfaced and addressed as quickly as possible during implementation.

Validation and Verification

Testing early and often is a central tenant of Agile software development. We want to surface potential problems early and address them as soon as possible. Functional, unit, and integration tests are written in the design phase and internal developers and testers will use these tests during the Testing phase to verify that each unit of code works on its own and as part of the whole.

After the tests are passed and output is validated in the Testing phase, the deployment phase is entered to verify End User Acceptance. If tests do not pass, the development phase is reentered. As outlined in the previous section, user acceptance is validated first among a small group of internal testers, and then gradually larger groups of students. Bugs or issues reported during user acceptance testing will be addressed in the next iteration.

Environments and Costs

Programming Environment

The program will be developed in C#, using Xamarin forms.

- Programming environment:
    - Windows 10
    - Visual Studio 2019

  - NETSTandard Library v2.03
  - NuGet package manager
  - SQLite-net-pcl v. 1.7.335
  - Xamarin.Forms v4.7.0.1142
  - Xamarin.Essentials v1.5.3.2
  - Xam.Plugins.Notifier v.3.0.0
- Target Operating System:
  - Android 9.0 Pie- API 28

Environment Costs

The application will run on user's mobile devices, minimizing environment costs. Additionally, the development environment relies on FOSS software tools. The SQLite database is stored directly on user's phones, and is free to use. The app will be distributed through Google Play Store for a one-time fee of $25.

Human Resource Requirements

- Project manager - 1
  - Performs requirements gathering, creates the project budget and timeline, and creates project requirements documents
  - Keeps stakeholders informed throughout the project
  - Ensures the project stays on schedule and within budget
- Project designer - 1
  - During the design phase, creates the wireframe
  - During additional iterations, provides input to design changes needed based on user feedback
- Developers - 2
  - Creates the UML diagram
  - Designs unit, functional, and integration tests
  - Code the program
  - Address bugs found in any phase of the project
- Senior Developer – 1
  - Reviews code for quality prior to entrance into deployment phase
  - Reviews unit, functional, and integration test results
- Quality assurance testers – 3
  - During the first deployment iteration, they focus on the project's functionality

Employee	Number In team	Hourly Rate	Estimated Hours	Estimated Cost
Project Manager	1	$75	480	$36,000
Designer	1	$35	24	$840
Developers	2	$50	160	$16,000
Senior Developer	1	$75	16	$1,200
QA testers	3	$25	60	$4,500
Total Estimated Cost				**$58,540**

Project Timeline

Iteration One

Phase	Milestone/Task	Deliverable	Description	Dates
Requirements gathering	Requirements gathering	Requirements Documentation	Meeting with customer and requirements review	5/31/2022 – 6/3/2022
Design	Project Kickoff	Requirements specification signed by all parties	Stakeholder meeting, including employees, to discuss and agree upon the previously created requirements documentation and kick off the project	6/3/2022
Design	UI design	Low fidelity wireframe and mockups	Mock up of the UI of the project	6/6/2022 – 6/7/2022
Design	UML creation	Class diagram	Creation of class UML diagram using the wireframe to guide development	6/8/2022
Design	Test plan	Test plans for: functional, unit, and integration testing	Developers design testing prior to writing any code	6/8/2022 – 6/10/2022
Development	Development environment set up	Development environment	Developers set up development environment	6/13/2022
Development	Code created	The program	Developers code the program	6/13/2022- 6/24/2022
Development	User guide	User guide	Developers create a user's guide	6/22/2022 – 6/24/2022
Testing	Test results	Testing	Developers run functional, integration, and unit tests	6/27/2022 – 6/29/2022
Testing	Code review	Code review results	Senior developer reviews the test results and the code	6/30/2022 – 7/1/2022
Deployment	QA results	QA documentation	Internal QA testers test functionality	7/3/2022 – 7/5/2022
Review	QA and test results	QA and test results for future iterations	The project manager reviews the QA team's findings to inform the requirements for the next iteration	7/6/2022

Outline for Future Iterations

Phase	Milestone/Task	Deliverable	Description	Dates
Requirements gathering	Requirements gathering	Requirements Documentation	Meeting with customer to review input from previous iteration, plan requirements for this iteration	TBD
Design	Standups/Scrum meeting	Requirements specification signed by all parties	Stakeholder meeting, including employees, to discuss and agree upon the previously created requirements documentation and kick off this iteration	TBD
Design	UI design	UI mockup	Designer creates a mockup of UI changes	TBD, may not be necessary
Design	Class diagram design	UML diagram	If major changes to project structure are needed, create an updated class diagram to reflect these	TBD, likely not necessary
Design	Test plan	Test plans for: functional, unit, and integration testing	Developers design testing prior to writing any code	TBD
Development	Development environment set up	Development environment	Developers set up development environment if there will be any changes	TBD, likely not necessary
Development	Code created	The program	Developers code required changes	TBD
Development	User guide	User guide	Developers update the user guide	TBD, may not be necessary
Testing	Test results	Testing	Developers run functional, integration, and unit tests	TBD
Testing	Code review	Code review results	Senior developer reviews the test results and the code	TBD
Deployment	UAT results	QA /UAT documentation and bug reports	User acceptance testing is performed, and bugs as well as user input are documented	TBD
Review	QA and test results	QA and test results for future iterations	The project manager reviews the UAT findings and bug reports to inform the	TBD

			requirements for the next iteration	

Repeat iterations as necessary during the life of the program