C868 – Software Capstone Project Summary

Task 2 – Section A

Capstone Proposal Project Name: Up To You

Student Name: Keegan Brewster

Table of Contents

Contents

Business Problem

The Customer

The customer is the Detroit City Council looking to increase awareness of local businesses and activities that the city has to offer. The city is looking to recover its public image by inspiring its residence and visitors to experience more of what the city has to offer. Visitors to the area do not know all the available option for dining and entertainment and frequently are plagued with making decisions on what to eat, what to do, or where to go. They are seeking a way to diversify visitors experiences and help them discover some of the less visited local attractions. The City Council wants to create a mobile application for residence and visitors to the city that will suggest fun activities or local restaurants near the user.

Business Case

The purpose of the mobile application "Up To You" is to provide a user friendly avenue to making decisions about recreational activities and food options in the nearby vicinity. Studies show that a surplus of choice is a negative thing for consumers and a person will become easily overwhelmed and make the same decisions because they are familiar with them or even give up on what they are deciding all together; this phenomenon is known as Analysis Paralysis and is prolific with a large majority of the modern-day population. "Up To You" provides a steadfast way to lead the user to more variety without the weight of decision making. Of course, making decisions on recreational activities and food options is a highly personal thing that is entirely specific to the person making the decision.

"Up To You" addresses these concerns in the following ways. Users will create preferences for certain activity types and cuisine choices. These preferences will weigh the potential outcome of the options and/or simply provide a "safe" option for the user. Different restaurants and activities nearby will be presented to the user based on the decision options selected. A final total of only one option will be presented to the user for selection out of all nearby items queried and filtered by the app, which will reduce choice anxiety and increase potential to experience new things.

Fulfillment

Up To You will be a mobile application, installable and usable on an Android based operating system, mainly phones. The application will be installed, and preferences will be stored locally on the device. Users will login to the application to save and load user specific preferences. The application will be broken up into different activities/screens that will hold a separate functionality in each section.

The first activity page is where users will be prompted to start their search for an activity or restaurant. There will be options for the user to designate the type of choice they are trying to make, after which they will define more specific details of the choice if they would like to.

The second page will be a follow up of the first, the application will take the predefined user preferences and the choices from the previous page to narrow down a decision. It will present the user one nearby restaurant or activity based on the preference type and will show this in a card format. The user will then be able to select the choice which best fits their current desires. This will be logged in the history.

A third activity will be accessible on the application which will be reachable from the main activity through a settings bar. This activity is like the initial login prompt for new users. It is a screen that will allow the user to change their preferences on items.

The final screen is where the user will be able to view decision history. This page displays each of the users' previous selections from using the application. History will be displayed in a card format and can be shared through a share functionality with others.

Existing Gaps

Up To You is not replacing any specific application currently in existence or being used but could more accurately be described as replacing the manual process of decision making for certain choices. As was stated previously, the process of decision making in the current day and age can be overwhelming. There are so many options out there for potential ways to spend your time, people can easily become afflicted by analysis paralysis. The primary gap for the manual process of this type of decision making is that it requires knowledge of all available options in the nearby area. Having and maintaining this knowledge is a monumental task today, which added to the difficulty of and pressure of decision making.

SDLC Methodology

The Software Development Lifecycle Methodology we will be using for the development and implementation of the applications first iteration of features will be the Waterfall methodology. This has been selected as the desired methodology for a few reasons. First of these reasons Is that due to the time constraints and personnel available to work on this project it is important to have all the details and deliverables of the application clearly laid out at the onset of the process. This will help prevent scope creep as the project progresses. This type of methodology is known as a predictive model and the waterfall methodology falls under this approach.

The waterfall methodology is broken up into different phases, each phase designed to provide a plan and base for the next phase. The phases can overlap somewhat, but for the most part each phase cannot start until the previous one has been completed. This rigid structure helps to keep the project focused and allows the team to provide a good estimation on the overall time for the project. First of the phases is the requirements phase, perhaps the most important phase of the project because it is where the idea is for the application gets fleshed out. This is where we will define the specific requirements and features for the software. We cannot go back and add features or requirements once this phase is complete so it's important to outline exactly what the application should be able to do. The result of this is a detailed list of each of the applications features and how they should work for the user.

By the end of the requirements phase there will be an official requirements document and project plan that details a schedule for the remainder of the phases. These items will be distributed to all applicable stakeholders of the project for review and accountability of project task timelines. The completion of these items leads into the system design phase. Some of the more significant and long-term impact decisions are made in the system design phase. None of the actual code for the program is written in this portion of the process, but it is where the software development language is selected. Along with identifying the coding language if it's not already known, this is where the platforms and specifications for running the application will be outlined. One can say that this is where the overall architecture of the program is decided upon. Database diagrams, UI design, wireframes, low and high-fidelity mockups, and a testing plan for the application are all produced in the design phase.

Next is the Implementation phase, where most of the actual coding work is done. The results of the previous phase are used here to start the building process by writing the necessary code. Portions of the application are built in units and broken up from the whole. This allows for developers to easily perform unit testing on each code section and confirm functionality before proceeding to the next phase. Although unit testing is performed by the developers in the implementation phase, the real bulk of testing is performed in the testing phase, which follows the implementation phase. During this phase, the requirements will be referenced heavily to identify whether all features for the application have been implemented and are working correctly. This is the portion of the waterfall process where bugs are identified and reported for fixing before the applications final release. It will include not only standard inhouse white box testing of the software, but also black box testing focused on user acceptance.

The culmination of the waterfall methodology we are implementing is the deployment phase, which will be where the application is installed and made available to the customer. For the purposes of this product, the APK file will be added to Google Play Store for discovery and user download. This will allow anyone utilizing the Google Play Store to download the application and install it on a supported device. Just because the application has been released and installed at this point, does not mean it is the completion of the waterfall methodology. In a way, there is never truly a completion of the process because the final phase of the methodology is the maintenance phase. This is more simply put, the support of the software, giving users the peace of mind about the products continued functionality by allowing for the report of issues and ensuring that the development team can address any issues that may arise causing loss or unintended hinderance of the applications use.

Deliverables

Here we will lay out all the deliverables related to the project. These are items that can be specifically provided to the customer and stakeholders throughout the entirety of the project. They will provide a unified image of what the current state of the project is, as well as being the source of truth for the project team as they move forward with the process. Deliverables will be broken down by type and sorted into sections based on the phase within the Waterfall method that they occur and will be received by applicable parties.

Provide information about what deliverables are related to your SDLC method. List and describe those deliverables. Also, include examples to help clarify what specific type of artifacts will qualify.

Project Deliverables

These consist of items that are part of the Project Manager’s realm of responsibilities.

- Project Schedule
  - Identify deliverables that will be provided throughout the project.
  - List of tasks for the project
  - Start and end dates for the tasks.
  - Dependencies for each task; including previous tasks that need to be accomplished to engage start each.
  - Project Calendar: the dates and timeline laid out for each project task and deliverable.
- Requirements Document

  - Features of the software application written out as verifiable statements.
  - Intended functionality of the software from a user perspective for comprehensive understanding by the developers.
  - Unambiguous as to the meaning and purpose of each requirement statement.
- Test Plans
  - The testing steps that the customer uses to perform validation.
  - Expected test results for each test.

Product Deliverables

Product Deliverables represents what is produced to deliver to the customer.

- Low & High-Fidelity Wireframes
  - Crude drafts of application appearance and basic flow through application screens.
  - More accurate/high detailed mockup of application interface with representation of functions for intended clickable UI objects.
- Entity Relationship Diagrams
  - Database diagrams for storage of background data of application.
  - Relationship diagrams for intended class objects and software application internal structure, including but not limited to, app screens, data objects, method structures.
- Prototype
  - Working basic application matching the low and/or high-fidelity mockups that has clickable UI elements and presents an interactable application.
- Final Software
  - Deployment of finished product with completed feature set and intended functionality of software application.
  - Support of software product through maintenance and user bug report support.

Implementation

The application will be implemented through the Google Play store. The APK will be posted on the Google Play store and be available for download and install on any supported Android based mobile device for free. Since the product is intended to be used by individual users visiting the city, it will be available for any qualifying user to download from the Play Store. A QR code will be displayed in advertising campaigns promoting the use of the app, which will direct users to the download link in the Play Store.

Validation and Verification

After implementing the software, the process continues with its validation and verification through requirements testing, multiple test scenarios, and user acceptance testing. A test group will be identified from various users and in-field testing will begin. The application will first be tested for each screen to ensure it functions as expected and fulfils the features outlined in the requirements document. Saving of user preferences will be tested and verification that preferences persist through closing and restart of the application in addition to power cycling the device.

Testing scenarios will be designed for the following steps which outline specific sets of preferences for different user profiles. The different user profiles will proceed through item selection within the application for multiple section including but not limited to; restaurants/food options, paid activities, and free activities. Each user set will proceed through item selection to verify preference related filtering is functioning appropriately. Additionally, location-based testing will need to be performed to verify accurate use of the proximity related features. Testers will use application in different identified locations for their user profile to verify location-based selections are accurate to specific users.

Environments and Costs

Programming Environment

Application development will be done using Android Studio environment on a Microsoft Windows machine. The coding language identified for use in the project is Java and the supported XML markup language for UI development in Android Studio. The application will utilize Room for SQLite as the applications data storage solution. Development with be done using an emulated Pixel 5 on API Level 33 with resolution 1080x2340, dp 393x851, RAM 4096, CPU Core 4. The minimum SDK requirement is 26 and the target SDK is 33.

Environment Costs

There are minimal environmental costs to the application, the database is stored locally on the device and is free to use. Google maps API is free for a significant number of licenses but will need to be paid for in a production environment. The pricing for use of Google Maps API will be the most expensive part of the application but can scale with the user base if there is an increase in application download and usage in the future. Pricing is based on usage, but Google provides a credit of $200 each month for businesses which applies to the usage of the API based on number of requests.

Human Resource Requirements

The most significant expense for the application in terms of human resources will be the cost for developers, but additional staff will be required for the success of the project. This includes the designers, project managers, software testing team, and customer care team. The project manager will be present during all aspects of the project, helping to facilitate timely deliverables and maintain customer expectations while keeping the whole team updated on current progress and next steps. This will cost roughly $30,000 for the course of the project. The designer will only spend time in the early portion of the project during the production of the high fidelity wireframe to create the look of the application, this will cost $2000. The development team will be involved throughout the majority project in the requirements phase, design phase, and obviously during the development portion of the project; this will cost $36,000 for the timeline of the project. The testing team will only be needed toward the end of the development phase and depending on the maintenance plan, through the foreseeable future, but for the purposes of the plan this will cost roughly $3000. The total cost for all human resources outlined in this plan will be $71,000.

Project Timeline

Phase	Milestone/Task	Deliverable	Description	Dates

Pre-development	Task 1	Requirements	Meeting with customer and procedure review	3/1/2023 – 3/17/2023
Design	Task 2	Low fidelity wireframe	Create the UI that relates the look and feel of the project	3/17/2023 – 3/20/2023
Design	Task 3	High fidelity wireframe	Create the UI that is more specific than the low fidelity wireframe and closer to the final design	3/17/2023 – 3/20/2023
System Design	Task 4	Entity Relationship Diagram	Create the logical matrix for data objects used within the applications database	3/20/2023 – 3/27/2023
Implementation	Task 5	Unit Test/Test Plan	Create unit test plan for specific code design elements. Finalize total test plan for complete acceptance testing.	3/27/2023 – 4/7/2023
Implementation	Task 6	Software Development	Initial creation of software application and implemented features in application.	3/31/2023 – 6/25/2023
Maintenance	Task 7	Develop and Persist service plan	Create maintenance plan for live application.	6/30/2023