C868 – Software Capstone Project Summary

Task 2 – Section A

Capstone Proposal Project Name: Android App for Fuel and Oil Change Entry

Student Name: Leonard Lutz

Table of Contents

Business Problem *1*

The Customer *1*

Business Case *2*

Fulfillment *2*

Existing Gaps *4*

SDLC Methodology *4*

Deliverables *6*

Project Deliverables *6*

Product Deliverables *7*

Implementation *7*

Validation and Verification *8*

Environments and Costs *8*

Programming Environment *8*

Environment Costs *9*

Human Resource Requirements *9*

Project Timeline *10*

References *11*

Business Problem

The Customer

The customer is JSE Electrical Contractors, Inc. (JSE), a well-respected provider of residential and commercial electrical services since 1981. Based in Waukesha, Wisconsin, the majority of their business is providing all of the electrical needs for new homes being built throughout southeastern Wisconsin and parts of northeastern Illinois. They also provide electrical services to new commercial and industrial customers in the same areas. Additionally, they provide electrical repair services to individuals or businesses with 24-hour emergency availability. This award-winning company has a full-time staff of over 150 full-time employees, including rough and finish electricians, design engineers, and office personnel.

JSE maintains a fleet of cargo vans, custom designed to carry all of the wire, outlets, fixtures, and everything else needed to complete each job. They also have delivery trucks to deliver any additional equipment or supplies necessary for larger jobs. Senior members of the company are provided with luxury cars to attend business meetings or visit job sites to determine the needs of the customer or to supervise the jobs as necessary.

The company maintains its own computer network running a variety of hardware and operating systems that they have determined provide all of their needs while remaining cost effective. They have their own data and web servers, and sufficient computing power for all of their required tasks. Some software has been commercially purchased, such as accounting, payroll, and inventory programs, but much of their software has been custom written to their own specifications. This includes the programs they use for job estimating, and specific software to help them maintain their fleet of vehicles.

Business Case

The owners of JSE, through past experience, have determined that maintaining their vehicles with regular oil changes and scheduled maintenance has proven to extend the vehicle's useful life, thus reducing the need to replace vehicles as often and saving the company substantial amounts of money. Because of this, the company commissioned a customized computer program to keep track of all fuel usage, oil changes, preventative maintenance, and expenses related to each vehicle. Their vehicle program provides extensive reporting capability that allows them to compare the operating costs and determine when it becomes wise to replace aging vehicles. The program also keeps track of the mileage and notifies the assigned driver of the vehicle (and the management) whenever an oil change or scheduled maintenance is due. The mileage is updated each time the vehicle has its fuel tank refilled. This information is entered by office personnel when receipts are submitted by the vehicle's driver.

Unfortunately, and too often, the paperwork is lost or not submitted, leaving gaps in the data and delayed services. There have also been occasions where the information was entered for the incorrect vehicle, causing data corruption. Therefore, JSE has decided that, rather than having the drivers submit paperwork to the office, and because JSE provides each of their employees with a company smart phone, they would have the drivers use an app on their phone to provide the required information immediately when purchasing fuel or services.

Fulfillment

The new application, which will run on the vehicle drivers' (company supplied) Android phones, will provide the drivers with the ability to enter all required information before they drive away from the fuel pump, or before they leave the facility providing the oil change.

When the employee is in the company office or in the warehouse picking up supplies, the app will detect and connect to the company WiFi and automatically download any changes to the Employee, Vehicle, Provider, Fuel, and OilChange tables, over the local connection using JDBC to interact with the company's MySQL database. A subset of this information is stored in an SQLite database on the employee's phone. For security, the database is not directly accessible over the internet[1]. However, updates can be sent ***to*** the office, via the company's WEB site, using the HTTP protocol and PHP programs that can access the database locally. Verification that the data was received and acted upon is returned to the phone the same way.

The application will not require a "Login" because it uses the phone number of the device it is running on to check the employee file and verify it is a company phone and to gather information about the user. Based on this information, the main screen displays the user's name and phone number it has found. Many of the company employees are assigned as the driver of a specific vehicle and that vehicle will be preselected in the vehicle list presented on the main screen. However, any company phone that has been assigned to an employee can be used to run this application, and any employee can select any vehicle in the list. Once the vehicle has been selected, the employee can press the appropriate button to add fuel or change oil.

The button press will bring up an entry screen where the user must enter all required information and select a provider from a list before pressing "SAVE". Upon verification of the data, the information will be saved to the phone's database and sent to the office. The user will then be returned to the main screen. The user can also return to the main screen at any time before saving by pressing the "CANCEL" button. The user may also use the main screen's drop-

[1] Because JSE no longer exists, and because people testing this application require access to a working database, a sample database has been installed on a website available to the author and has been made accessible over the internet.

down menu to view historical data on all of the fuel fillups or oil changes for any vehicle. Also from the drop-down menu, the user can select to request a database update or close the app.

It is imperative that data integrity be preserved, so it was determined that any changes to data, or deletions of records, must be performed by personnel trained in database operations. Therefore, editing and deletions will only be allowed by office personnel that have received such training. These procedures must be done using the existing software on office computers, so they will not be included in this phone application.

Existing Gaps

The original process requires all fuel fillups and oil changes to be entered by office personnel when (and if) the receipts with all the information are actually turned in to the office. Receipts are still required to be turned in to the office, but this new app will provide more timely updates which can help the company monitor their fleet of vehicles in nearer to real time, and also free up office personnel to attend to more pressing matters.

SDLC Methodology

Recalling teachings from *C188 – Software Engineering*, we learned (UCertify, 2020):

> The waterfall model can work reasonably well if all the following assumptions are satisfied:
>
> - The requirements are precisely known in advance.
> - The requirements include no unresolved high-risk items.
> - The requirements won't change much during development.
> - The team has previous experience with similar projects so that they know what's involved in building the application.

- There's enough time to do everything sequentially.

This project certainly meets all of the criteria, so this is the model we will be following for the development of this application. The waterfall method includes the stages: Requirements, Design, Implementation, Verification, Deployment, and Maintenance. For each of these stages, the work and the personnel involved will be:

- Requirements
  - The project manager will meet with all stakeholders to define and agree to all specifications and requirements, as well as an acceptable timeline to complete the project. All personnel that will be involved in the project, from both parties, will also be determined.
- Design
  - The design team will meet with specified members of the customer's team to discuss requirements, data involved, and how the users will expect the program to look. Concept drawings of all expected screens will be produced and agreed upon.
- Implementation
  - The programming team will create the application based on the design and requirements documentation. Unit testing criteria will be updated and continuously run throughout this process.
- Verification
  - Customer personnel will be provided with beta-test versions of the application to test and verify that it meets all customer requirements. Any issues found will be fully documented and corrected.

- Deployment
    - The application will be delivered to the customer for installation on their employee phones a few at a time, while monitoring usage and accuracy to ensure that the application is working as expected.
- Maintenance
    - Any irregularities (bugs) will be documented and corrected, and the application updated as quickly as possible. Feedback and customer satisfaction will be monitored and acted upon appropriately, including evaluating if any issues found will be considered as bugs or possible future enhancements.

Deliverables

Each phase of the project will produce specific items that might be provided by either party. These may include specific documentation from either party, agreements signed by both parties, information required by one party or the other, etc., and of course, the application itself. The main deliverables for this project are:

Project Deliverables

- Requirements Document
- Project Scope
- Project Charter
- Project Schedule
- Status Reports
- Test Plan

- Entity Relationship Diagram – This would normally be a “Product Deliverable” from the provider to the customer. In this case, because this application is sending updates to an already existing database, the customer will need to deliver the database information to the provider.

Product Deliverables

- Proposed Screen Designs
- Application
- Documentation

Implementation

Due to the fact that this application is simply providing a remote method to enter limited data into a working database that is maintained by a complete set of programs supporting all aspects of the database, implementation will not require any shutdown for the customer. In addition, the database is not being changed and, for security, only portions of some of the files will be copied to the user’s phones. No sensitive information, i.e., home addresses or social security numbers will be transferred.

The initial roll-out will begin with the application being installed on 10-15 employee phones. Any issues found will be resolved and the app will be updated on the currently deployed phones. Customer callbacks and user satisfaction will be monitored, and feedback will be documented to watch for any issues or possible enhancements. After there have been no issues reported for at least a week, customer representatives and members of the development team will begin the validation and verification process.

The development team will train selected office personnel on how to install the application on the employee's work phone and to perform the initial transfer of data from the company server to the phone. All of the steps to perform these tasks will also be documented in the application's *User Guide*, as will all information required for normal use of the app.

Validation and Verification

The validation process will request stakeholders to confirm that the application is working as expected and all requirements have been met as specified in the requirements document. Any new requirements that were not originally agreed upon will be considered enhancement requests that will incur additional costs.

Throughout the development process, all code will be constantly tested, using the agreed upon test plan, to make sure that the application continues to meet all documented requirements, performs as designed and expected, and meets all functional, non-functional, and business requirements as well. Testing will be automated as much as possible. However, with the application designed to directly access tables in the company's MySQL database, while it updates a local database on the phone, and sends updates over the internet to the company's web server for processing through PHP programs, significant testing will need to be performed manually.

Environments and Costs

Programming Environment

The provider already owns all hardware and software required to complete this project, including computers running Windows 10 or higher, Java 8 or higher, and Android Studio 4 or

higher. The provider also has the required access to a local MySQL database server version 5 or higher and a web server that supports PHP 5.4 or higher.

Environment Costs

The company already possesses all of the required phones, computers, software, licenses, servers, etc., required for this application. Therefore, they will not incur any new or additional environment costs other than the purchase of this app.

Human Resource Requirements

The software development team will be the predominant resource required for this application. Their involvement will be required throughout the project and will likely be responsible for about 80% of the budget. This leaves about 15% of the budget for project management and the remaining 5% for administrative activities.

Project Timeline

Phase	Milestone	Deliverables	Description	Dates
Requirements	Task 1	Requirements Document Project Scope Project Charter Project Schedule Test Plan Database Schema	Meetings with customer team about procedures and test plan	4/12/2021 – 4/16/2021
Design	Task 2	Low fidelity concept High fidelity mockup	Create the UI that displays the look and feel of the project	4/19/2021 – 4/23/2021
Implementation	Task 3	Working Android Application with Database Connectivity	Develop the application and middleware PHP programs	4/26/2021 – 5/7/2021
Verification	Task 4	Test Results Bug Reports Acceptance Document	Verify the app is fully functional and working as expected	5/10/2021 – 05/14/21
Deployment	Task 5	Fully Tested Application	Application installed on user's system and employee phones	5/17/2021 – 5/21/2021
Maintenance	Ongoing	Customer support	Provide support as needed	5/21/2021 – 5/20/2022

References

UCertify. (2020). *Beginning Software Engineering*. Retrieved Jan 10, 2021, from ucertify.com:

https://wgu.ucertify.com/?func=ebook&chapter_no=14#02TfW