

C868 – Software Capstone Project Summary

Task 2 – Section A



Capstone Proposal Project Name: WeSchedule – A Customer / Appointment Scheduler

Student Name: James Blankenship

Table of Contents

Business Problem	3
The Customer	3
Business Case.....	3
Fulfillment.....	4
Existing Gaps.....	5
SDLC Methodology.....	5
Deliverables.....	7
Project Deliverables	7
Product Deliverables	9
Implementation.....	9
Validation and Verification	9
Environments and Costs.....	10
Programming Environment.....	10
Environment Costs	10
Human Resource Requirements	10
Project Timeline	11

Business Problem

The Customer

The customer is Northwest Schedules, a global consulting organization with main offices in Phoenix, Arizona; White Plains, New York; Montreal, Canada; and London, England. Northwest Schedules serves around 25,000 customers a year with scheduling software. The consulting organization currently has 25 employed consultants and plans to expand its business sometime in the future. Their current system is outdated and causes customers and contacts frustration due to bugs and errors. They are looking to replace their current system so their customers can have a better overall experience, resulting in more favorable service. Northwest Schedules also wants the new system to be scalable to expand its organization when needed.

Business Case

The proposed software solution is a complete customer appointment management system designed to meet the needs of Northwest Schedules and other scheduling organizations. It will aid in tracking customer appointments with a particular contact and managing customer and appointment records. Northwest Schedules has many records, and with that, their current system is not designed to work with their requirements. The system in its current state frustrates users and causes customer record loss and contacts to miss appointments. The proposed solution will be capable of addressing these errors by using updated programming features in an all-in-one Java application. The application will use a MySQL relational database with normalized tables, preventing inefficient queries to the database. The proposed solution will be capable of handling the current load of Northwest Schedules and more.

Fulfillment

WeSchedule will be a Java-based native desktop customer appointment tracking application used by approved technical staff at Northwest Schedules. All functionality will be locked behind a secure sign-in form to protect the customer and contact information further. The form will employ preventative measures to protect the database from SQL injection attacks. New contact creation will be disabled by default. Three different report sections will be included in the application, allowing the user to manage customers and appointments more reliably and efficiently.

The first section will allow the management of customer appointments by type and month. The application will have the ability to display a list of appointments by type and month in a table format, calculating the total number of appointments per category. This report information can help the organization better analyze its application. The modify appointment interface can easily update information on customer appointments.

The second section will be designed to manage contact schedules. All contacts will be listed with a schedule of appointments to better contact's time management. Appointment details, such as appointment id, title, type, description, start time, end time, and customer id, will be available on the report under the contact's name.

The last section will include the management of contact appointment hours. Appointments will have start and end times and be linked to a particular contact and customer. The application will calculate and display the total hours of appointments each contact has scheduled in an easy-to-read format. Date validations will ensure that neither the contact nor the customer has overlapping appointments. It will also be impossible to create appointments outside of business hours.

The application will use an industry-standard MySQL relational database with normalized tables. The database will be designed to be as efficient as possible to ensure the organization can perform as needed. The application will protect the data in the database from any malicious attacks, leaving the data secure and untouched.

Existing Gaps

Northwest Schedules is experiencing a tremendous amount of problems with its current system. Users and technical staff frequently complain about scheduling errors and missed notifications due to bugs in the system. Many contacts employed by the organization miss their appointments, which upsets customers and creates a hectic work environment.

The current system in use by Northwest Schedules has not been updated for two years. Due to this, some programming mechanisms have been depreciated and no longer work. The system needs a total re-model using today's best practices to ensure the organization does not continue to lose customers.

The problem of tracking Northwest Schedules' customer appointments requires precision and organization. A native up-to-date Java application to manage the organizations' data in one location would eliminate most, if not all, of the difficulties the customers and clients face. An application with a centralized MySQL relational database with a friendly user interface would allow the technical staff to create, update, delete, and search customer and appointment records with ease in a secure and efficient way, lowering the rate of potential errors and reducing memory usage.

SDLC Methodology

For delivering the first phase of features for the WeSchedule application, we have chosen the Agile methodology for developing software applications. This methodology is categorized as an iterative approach to developing software, where the project's features and requirements evolve throughout the development cycle. Once prototype testing occurs, users may ask for something to be changed or new features added. Using Agile would allow requirements to change throughout the development life cycle based on the end user's needs, creating a more sustainable application. The Agile method is well suited for a project like WeSchedule since the problems the consulting organization is facing are well understood but may change after testing due to user feedback.

The Agile methodology begins with the concept phase, where the product owners seek to determine the project's scope. The product owner will then discuss with the client the project's essential requirements and prepare the relevant documentation to outline them. Lastly, the product owner estimates the cost and time of the project, which will help decide whether the project is feasible. The main deliverables in this phase are the project vision, sprint backlog, and dynamic product backlog documents that are distributed to the project stakeholders.

The concept phase is followed by the inception phase, where the product owners find and employ skilled developers and designers for the project. The product owners provide the designers and developers with all the resources and necessary tools to start designing the project. The next step is to create a user interface mock-up and build the project architecture. The main deliverables in this phase are the support documentation, low-fidelity wire-frame, and high fidelity prototype to help with design decisions.

The next phase is the iteration phase, arguably the most prolonged phase of all, where the development team works with user experience designers to combine all of the product requirements and customer feedback into code. This phase focuses on turning the design documentation into software by the end of the sprint. The produced software goes through various revisions before it has reached its final version to ensure a good quality product. One of the essential characteristics of this phase is continual testing to ensure quality code. By the end of the sprint iterations, the product will be a fully functional solution ready to go into testing. The main deliverable in this phase is the finished software product.

The iteration phase is followed by the testing phase, which focuses on documentation development, quality assurance testing, and releasing the final iteration into production. This stage's time is mainly spent analyzing the application to ensure it is bug-free and compatible with all written sprint backlog features. After all the testing has been completed, the software is ready to go live. The main deliverables in this phase are a testing plan and user documentation.

Once the testing results have been ruled successful, the production phase can begin. This stage fully deploys the software and makes it available to customers. Simultaneously, the software teams will go into the maintenance phase to continually ensure the system continues to run smoothly and is free of potential errors.

Finally, in the review phase, the product owner and team members review the completed project. They evaluate the progress to date and document the highlights of the project and the roadblocks that occurred. This phase is crucial because it allows the team to understand what went right and wrong throughout the project to execute the next sprint's tasks better. After this phase, the life cycle starts a brand new iteration. The product owner may ask stakeholders for any feedback that will help the next iteration.

Deliverables

The Agile methodology of developing software has teams that produce as little documentation as possible. The documentation that is produced can be divided into project deliverables under management's authority and product deliverables that illustrate the software product to be delivered to the customer. We will have a deeper look into the project deliverables in the next section.

Project Deliverables

- Project Vision
 - Description of what the organization or project team wants to accomplish or obtain upon completion of the project.
 - Needs to be a clear and concise description of the big picture.
- Sprint Backlog
 - List of prioritized work items the team plans to complete during the sprint duration.
 - Work items are usually derived from the product backlog during sprint planning sessions.
 - Needs to be clear and concise in who is responsible for what to prevent scope creep from happening.

- Product Backlog
 - Prioritized list of work items for the software development team that is obtained from the roadmap and its requirements.
 - The most important work items are shown at the top of the backlog, so the team understands what to deliver first.
- Support Documentation
 - Training materials specific to staff, trouble-shooting guides, templates, examples, etc.
 - Needs to be understandable and unambiguous.
- Low-Fidelity Wireframe
 - A basic design that outlines blueprints for the user interface screens.
 - Designed to display the functional aspects of the application consist of and the flow of the application.
 - Does not need to consist of a high level of detail but needs to display the main navigation routes throughout the screens and how the user can interact with the application.
- High-Fidelity Prototype
 - Based on the finalized wireframe, the prototype is created with greater detail.
 - Consists of most of the major features and design elements to focus on the look and feel of the application.
- Testing Plan
 - Includes unit testing, integration testing, system testing, and acceptance testing results to ensure the application has the correct functionality. The testing will be manually done and produced in a structured output.
- User Documentation
 - Content that informs the user in order to be more successful with the application.
 - Also known as instruction manuals, user guides, or user manuals.

Product Deliverables

- Fully developed application that meets all functionality in the requirements documents.
- Flow and navigation that mimics the wireframe.
- User interface that resembles the one shown in the high-fidelity prototype.
- Secure access through a log-in form.

Implementation

The implementation of the project is expected to occur while the organization is closed, minimizing the risk of disruptions. By the time of implementation, the client has participated in the design of the project through acceptance testing and usability testing and has a solid understanding of what the end result will be.

Once the application is deployed into the production environment, the organization's software team can create the needed accounts for the staff members. After the accounts are created, the application will be ready to use by Northwest Schedules.

Validation and Verification

A detailed testing plan will be created to ensure the application meets the needs of the customers as stated in the requirements documents. Quality assurance and testing are an essential part of the Agile methodology, driving high-quality software products in our company.

The software engineers at our company will create unit tests alongside code to verify that the functionality of the methods in the application works correctly. The unit test breaks the program down into manageable parts so the code can be tested at a lower level. The tests will be run manually and will be maintained for further testing.

Acceptance and user testing will seek to ensure the customer's needs are met. The customer will be involved in testing the application to recommend any changes, add new features, and validate functionality.

Environments and Costs

Programming Environment

The application will be developed chiefly using Java, but several files use FXML to build the user interface. The database back-end will be a MySQL relational database capable of efficient queries and normalization. The application will be available for download on Windows 10 computers, automatically connecting to the database upon being loaded.

Environment Costs

The cost of running the application in a production environment will be modest. The database plan has different packages based on the desired user count, and with roughly 25,000 users, the cost will be \$7/month. If the database is nearing full, plans can be upgraded and customized to the organization's needs—the total yearly cost of hosting amounts to \$84.

Human Resource Costs/Requirements

The project requires a project manager, a user interface designer, a user experience designer, one software engineer, and a quality assurance specialist. The project manager will oversee the project throughout its life-cycle, plan activities are resources, organize and motivate the project team, maintain time management, estimate costs and budget, manage project risks, and more. His estimated activity amounts to 50 hours, at a rate of \$70/hour, totaling \$3,500. The designers charge \$50/hour with their activity is mainly confined to the inception phase. The estimated time to design the interfaces amounts to 35 hours for an application of this size, totaling \$1,750. The software engineer is expected to account for the majority of the overall development costs, with most of the work being done in the iteration phase. Their rate is \$75/hour with an expected development time of 80 hours, totaling \$6,000. Finally, the quality assurance specialist operates during the testing phase and requires 20 hours at a rate of \$40/hour, totaling \$800. The total cost of the human resources is estimated to be:

Resource	Rate * Time	Total
Project Manager	\$70/h * 50h	\$3,500
Designers	\$50/h * 35h * 2	\$3,500
Software Engineer	\$75/h * 80h	\$6,000
Quality Assurance Specialist	\$40/h * 20h	\$800
TOTAL Human Resource Cost		\$13,800

Project Timeline

Phase	Milestone/Task	Deliverable	Description	Dates
Concept	Discuss and finalize requirements	Project vision, sprint backlog, product backlog	Meetings with stakeholders to assemble and prioritize product requirements	6/1/22 - 6/4/22
Inception	Create and test low fidelity wireframe and high-fidelity prototype	Low-fidelity wireframe and high-fidelity prototype	Designers will create a low-fidelity wireframe to display navigation and flow and a high-fidelity prototype that will display the final look of the application.	6/5/22 - 6/12/22
Inception	Create support documents	Support documents	Project managers or other management will create support documents to help staff and users to be more informed	6/5/22 - 6/8/22

Iteration	Create a functional application with unit tests	Specific version of the application	Engineers will implement the requirements and mimic the prototype design to create a fully functional Java application according to specifications. All unit tests will run manually without errors.	6/13/22 - 6/27/22
Testing	Create overall testing plan and user documents	Testing Plan	Project manager and engineers will create a detailed plan for unit and acceptance testing, and documents to help staff and customers.	6/30/22 - 7/5/22
Testing	Create and execute acceptance tests	Customer executes acceptance tests till satisfaction	Under authority, customers will run an acceptance test to verify that the application meets their requirements.	7/6/22 - 7/8/22
Production	Prepare the deployment environment and deploy application	Application ready to be cast off to the customer	Deploy application after final checks are successful	7/11/22 - 7/12/22
Review	Review the project and how it went.	Sprint Summary	Discuss what went right, what went wrong, and what can be done better next time	7/13/22 - 7/15/22