C868 – Software Capstone Project Summary

Task 2 – Section A

Capstone Proposal Project Name: TechGopher Scheduler - A Client Scheduling Application

Student Name: Evan A Smith

TechGopher Scheduler - A Client Scheduling Application

Table of Contents

Business Problem....................4

The Customer....................4

Business Case....................6

Fulfillment....................7

Existing Gaps....................8

SDLC Methodology.................... 10

Deliverables.................... 12

Project Deliverables....................13

Product Deliverables.................... 14

Implementation....................14

Validation and Verification....................16

Testing....................16

Environments and Costs.................... 20

Programming Environment Overview....................20

Environment Costs Overview 22

Hardware Costs and Requirements.. 22

Human Resource Costs and Requirements 22

Project Timeline 24

TechGopher Scheduler - A Client Scheduling Application

Business Problem

The Customer

The Customer is Tech Gopher LLC., a consumer electronics repair company originating from Minneapolis, Minnesota in the history Seward Neighborhood, that has recently expanded to several new locations, and is now serving customers in a tri-state area at 12 separate locations. Tech Gopher specializes in repairing a wide range of consumer electronics, including smartphones, laptops, tablets, gaming consoles, and smart home devices. The company is dedicated to providing fast, affordable, and high-quality repair services, ensuring customer satisfaction and device longevity. The business is supported by a team of store managers, technicians, customer service representatives, and an IT and operations team. Store managers oversee daily operations at each location, ensuring efficient workflow and quality control. Technicians handle the repairs and diagnostics, while customer service representatives assist with inquiries, scheduling, and post-repair support. The IT and operations team maintains the company's technological infrastructure, ensuring smooth scheduling, database management, and customer interactions.

TechGopher Scheduler - A Client Scheduling Application

Ensuring high quality customer service experiences is critical to the continued success of TechGopher, although recent expansion to new locations and the training of new employees have likely contributed to a noted uptick in customer complaints that the location was disorganized, or did not know the purpose of their appointment in advance, or did not record the incoming. By all accounts, the existing systems are outdated. Customer reservations are primarily taken over the phone and recorded in a monthly planner. These records are later transferred to multiple Microsoft Excel spreadsheets, which often become disorganized and dispersed across different areas of the business. It has become increasingly urgent to implement a more advanced technological solution to support both current and future operations at TechGopher LLC.

In the short term, Tech Gopher aims to implement an improved appointment scheduling system to streamline customer bookings and reduce wait times. The company is also focused on enhancing customer support features, such as automated notifications and real-time repair tracking, while expanding marketing efforts to increase brand awareness in new service areas. In the long term, Tech Gopher plans to open additional locations to serve a broader customer base and introduce subscription-based maintenance plans for recurring customer support. The company also seeks to develop a nationwide mail-in repair service, allowing customers to send in their devices for professional repairs, and invest in AI-driven diagnostics and automation tools to improve repair efficiency and accuracy.

TechGopher Scheduler - A Client Scheduling Application

With a growing demand for electronics repair services, Tech Gopher has experienced consistent expansion. By leveraging advanced scheduling and management solutions, the company anticipates a 20% increase in customer volume over the next three years. Tech Gopher is committed to scaling efficiently while maintaining its high standards of service, ensuring that customers continue to receive the best repair solutions available.

Business Case

Tech Gopher LLC currently faces challenges in managing client appointments efficiently across its multiple locations. The existing scheduling process is prone to inconsistencies, leading to missed appointments, double bookings, and difficulty in tracking client interactions. Additionally, there is no streamlined method to document essential details such as the device manufacturer, the purpose of the visit, or the client's contact information in an organized and easily accessible manner. Without a centralized system, technicians and customer service representatives struggle to quickly retrieve customer history, leading to delays and inefficiencies in service delivery.

The scheduling application will address these issues by providing a centralized, user-friendly system for tracking client appointments. The application will allow employees to schedule and manage appointments while attaching detailed notes about the client's device, including the manufacturer and the reason for the visit. Additionally, it will maintain an up-to-date client database, ensuring that customer contact details are easily accessible for follow-ups, notifications, and service reminders.

Through its automated scheduling features, the application will help prevent double bookings and reduce scheduling conflicts. Clients will receive confirmation and reminder notifications, minimizing the risk of missed appointments. The system will also provide technicians with quick access to customer repair history, allowing them to prepare for appointments in advance and offer more personalized service.

By integrating these features, the scheduling application will streamline the appointment management process, enhance customer communication, and improve overall efficiency. This will ultimately lead to better customer satisfaction, reduced administrative workload, and increased business productivity for Tech Gopher LLC.

Fulfillment

The TechGopher Scheduler App will be a windows-based application which allows users to access the centrally-hosted client_schedule database from any company PC. The database is structured to enable robust customer contact capture, and facilitate detailed appointment information and scheduling. All functionality will be behind a secure sign-in. New account creation will be disabled by default pending executive outline of the company's IT Security policy, and approval of the app's rollout. There will be four main sections to enable the logistics of managing appointments effectively and reliably.

The application will provide functionality to list customers and quickly search for a specific customer by name. This screen will also show upcoming appointments for the selected customer and offer the ability to schedule new ones. Additionally, the application will allow for easy updates to customer information. This same Appointments screen will display customer details, including contact address, phone number, and appointment purpose, alongside a list of upcoming appointments. Users will also have the ability to add new appointments and modify customer information as needed.

Lastly, the application will have some useful reports to view trends in the appointment data. There will be graphs to display the number of appointments by date, customer, and by time range to assess more or less busy periods for making future data-driven scheduling decisions.

The application will be backed by an industry standard mySQL database, which will enable possible future integrations with other software solutions such as systems for keeping detailed records about customers and their appointment history. In addition, it will also be possible to transfer existing customers, staff, and appointment records from spreadsheets to the new database using custom data import scripts.

Existing Gaps

Tech Gopher LLC currently relies on a manual appointment scheduling system that includes a combination of spreadsheets, phone logs, and handwritten notes to track client appointments and repair details. This process is outdated and inefficient, leading to frequent scheduling conflicts, lost customer information, and difficulty in managing service histories across multiple locations. Store managers and customer service representatives often struggle to maintain consistency, especially when dealing with last-minute appointment changes or high customer volume. Additionally, technicians lack a streamlined way to access client details and repair history, resulting in delays and miscommunication.

The existing system also lacks automated reminders for customers, leading to a higher rate of missed appointments. Since appointment records are often fragmented across different locations, it becomes challenging to track repeat customers, manage warranties, and provide personalized service. Without a centralized database, employees spend excessive time searching for information, which reduces overall efficiency and negatively impacts customer satisfaction.

The new scheduling application will replace this antiquated system with an automated, centralized solution that will provide real-time appointment tracking, integrated customer data management, and automated notifications. By eliminating manual record-keeping, Tech Gopher will reduce errors, improve efficiency, and enhance customer communication. The new system will ensure accurate scheduling, streamlined technician workflows, and better overall service management, making it a vital upgrade for the company's expanding operations.

The scheduling application will include built-in cancellation and rescheduling management to efficiently handle last-minute changes. When a customer needs to cancel or reschedule an appointment, they will have the option to do so by calling the store, where staff can quickly update the system.The challenge of accurately tracking customer appointments and related records demands a high level of precision and organization. Computerizing this process by managing all data in a single system would greatly mitigate these difficulties. A centralized, database-backed application with a user-friendly interface would enable staff to efficiently enter, update, and search records, ultimately improving workflow efficiency and significantly reducing the risk of manual errors.

SDLC Methodology

Because of the clearly defined business need for a more robust appointment management system, and the significant increment of effect computerization alone will impart, the predictive approach of the Waterfall Software Development Life Cycle has been chosen. With a predetermined set of requirements, development follows a structured sequence of stages, each with its own deliverables.

The Waterfall method is particularly well-suited for the TechGopher Scheduler project, as the business challenges are well understood, and the necessary features for managing customer contacts and appointments can be carefully planned in collaboration with the client.

The Waterfall method starts with the requirements phase, which focuses on understanding the client's current situation and needs. During this stage, a comprehensive set of features is outlined, and the desired functionality of the software solution is detailed. The primary deliverable from this phase is a requirements document shared with all stakeholders. Additionally, a project schedule outlining the various stages of development, along with estimated timelines, is provided at this point.

The system design phase follows the requirements phase, during which the project team constructs the system's framework. While no coding takes place in this stage, key technical specifications—such as programming languages, database structures, deployment strategies, and testing approaches—are determined. The primary deliverables include a low-fidelity wireframe to outline the application's flow, a high-fidelity prototype showcasing the UI design, a database diagram mapping data structure and relationships, and a comprehensive testing plan to ensure full functionality.

Next is the implementation phase, where the actual development of the software takes place. Using the outputs from the previous stages, programmers and designers begin building the application.

This phase is subdivided into smaller stages, allowing the team to periodically review progress, align efforts, and ensure adherence to the plan. The final output of this phase is a functional software product that meets all the specifications outlined in the requirements document.

Following implementation, the testing phase focuses on identifying and addressing any deviations from the agreed-upon requirements. Quality assurance teams rigorously test the application, documenting any bugs or errors encountered. The engineering team then promptly resolves these issues to ensure a polished and reliable product.

Once all known issues have been addressed, the delivery or deployment phase begins. This step transitions the application into a live production environment, making it ready for customer use. The primary deliverable at this stage is a fully operational software application.

Finally, the maintenance phase ensures continued support for the customer. This includes fixing newly discovered bugs, releasing patches, and implementing updates based on customer requests. A maintenance contract will outline the terms of ongoing support services to ensure the application remains functional and up to date.By following the Waterfall methodology, the development of TechGopher's Appointment Scheduling App will be systematic and structured, ensuring a stable, well-tested final product.

Deliverables

As outlined above, the Waterfall method follows a structured approach with clearly defined deliverables at each stage of the project lifecycle. These deliverables can be categorized into two main types: project deliverables, which fall under the responsibility of the Project Manager, and product deliverables, which encompass the actual software product delivered to the customer. The following section will provide a detailed breakdown of both categories.

Project Deliverables

- **Requirements Document**: Provides a detailed outline of the software application's features and intended functionality, capturing the development team’s understanding of the customer’s needs, system constraints, and specifications. Must be written clearly and unambiguously to serve as a reference throughout all project stages. Should include key details such as user interfaces, functional capabilities, performance expectations, data structures, and any constraints or limitations.
- **Project Schedule**: A comprehensive timeline that maps out the main stages and substages of the project, along with their respective deliverables and estimated timeframes. Should outline tasks, milestones, and brief descriptions of each to ensure clarity and prevent ambiguity.
- **Low-Fidelity Wireframe**: Provides mockups of the entire application, showcasing the core flow and primary functional aspects. Does not require detailed design elements but should clearly depict navigation components and user interactions.
- **High-Fidelity Design Prototype**: Builds upon the wireframe to establish the application's look and feel, incorporating complete theme elements such as typography, color schemes, and branding.
- **Database Diagram**: Serves as the blueprint for the database schema, detailing structures, data types, and relationships between tables. Offers a comprehensive representation of real-world entities such as customers, appointments, and services.

Testing Plan

- **Unit Testing**: A suite of automated unit tests designed to validate individual code components. These tests leverage white-box testing methods to ensure proper functionality.
- **Functional Testing**: A structured test plan executed by QA specialists, following predefined steps to verify the application's end-to-end functionality. Each test step includes expected inputs and outputs to confirm accuracy.

Product Deliverables

- Fully Functional Database with a custom schema that aligns with the database diagram and supports the appointment application's data storage needs.
- Complete Application that implements all features outlined in the requirements document.
- Navigation and User Flow consistent with the structure defined in the wireframe.
- Graphical User Interface (GUI) that matches the high-fidelity design prototype.
- Secure Admin Access for designated staff members to manage and maintain the system.
- Initial Data Import Support to streamline the transition and facilitate immediate use of the application.

Implementation

During the testing phase, the **Project Manager** will ensure that selected TechGopher staff receive adequate training to facilitate the adoption of the new software. This preparation will enable the customer to successfully conduct User Acceptance Testing (UAT) before approving deployment. Additionally, the software team will develop custom data transformation and import scripts to populate the new database with initial values, including customer details, phone numbers, home addresses, and appointments. This will streamline the transition and enhance the immediate usability of the application, coordinated with the personnel leading the process. Once the application is deployed to the production environment, the software team will create admin accounts for designated staff members.

Outcome: Validate that the software is aligned with user needs, and any remaining issues are addressed before full deployment.

Personnel Roles: Several key personnel will be involved in the implementation process, ensuring the project is deployed effectively:

Project Manager: Responsible for overseeing the implementation process, coordinating resources, managing timelines, and ensuring communication between teams.

Development Team: Responsible for deploying the application, fixing bugs, and making necessary adjustments based on feedback from the validation and verification processes.

Quality Assurance (QA) Team: Conducts testing throughout the phases, validating the functionality, stability, and performance of the application.

Product Owner: Represents Tech Gopher LLC.'s interests, providing feedback on the software's functionality and guiding development based on user needs.

End-Users (Technicians, Customer Service Reps, etc.): Actively participate in validation and verification by providing feedback on the application's usability and functionality.

IT Support Team: Assists with the technical aspects of deployment, including system integration and troubleshooting during the implementation phases.

Training Team (if applicable): Provide training to the users (technicians, administrative staff) during the implementation phases, ensuring that they are comfortable with the software.

Timing and Customer Needs:

The timing for the rollout will be carefully planned to ensure minimal disruption to Tech Gopher’s daily operations. The deployment will occur after business hours when possible, and support will be available to address any issues quickly. By conducting the implementation in waves, the project allows enough time to address issues as they arise, ensuring that the final deployment is as seamless as possible.

Wave 1: Expected to take place during a 1-2 week period to allow for bug identification and initial validation.

Wave 2: Scheduled for 2-3 weeks after incorporating feedback from Wave 1, expanding to a wider audience.

Wave 3: Full deployment is planned for 1-2 weeks following the completion of Wave 2, ensuring all issues are addressed.

By the end of the full rollout, the scheduling application will be fully integrated into Tech Gopher’s operations, with all locations using it for customer management and appointment scheduling.

Validation and Verification

To ensure that the TechGopher Appointment Scheduling App functions correctly and meets the customer's needs, a combination of testing methods will be employed throughout development. The primary goal is to verify that the application provides the required functionality, maintains data integrity, and ensures application security.

Testing Methods

Unit Testing: Performed by Developers using JUnit during the implementation phase.

Objective: Validate individual components of the software to ensure they function as expected.

Scope: Testing core functionalities, such as creating, updating, and deleting appointments. Validating the Appointment class using built-in JUnit libraries. Ensuring SQL queries correctly manipulate the database.

Integration Testing: Performed by developers after unit testing is completed.

Objective: Verify that different modules work together as expected.

Scope: Ensuring that appointment data is correctly retrieved and stored in the SQL database. Validating that the login system properly authenticates users and restricts unauthorized access. Confirming that appointment changes reflect accurately in the system.

System Testing: Performed by Developers, testers, and QA engineers before deployment.

Objective: Test the entire system to ensure it meets functional and non-functional requirements.

Scope: Verifying that the scheduling system supports all required operations. Checking UI elements for usability and proper workflow. Ensuring database interactions are secure and efficient.

User Acceptance Testing (UAT): Performed by the customer before taking ownership of the application.

TechGopher Scheduler - A Client Scheduling Application

Objective: Confirm that the application meets business requirements and is ready for deployment.

Scope: The customer will test the application in a real-world environment. Evaluate appointment scheduling, modifications, and security features. Ensure that user authentication works as expected.

To prove that the scheduling application functions sufficiently well to meet Tech Gopher LLC.'s needs, we will utilize several methods to ensure the software delivers all required functionality. These methods will include unit testing, integration testing, system testing, user acceptance testing (UAT), and regression testing. The first step is to validate whether the software delivers all required functionalities. This will be done through Acceptance Criteria defined in the user stories during the development phase. Each user story will have clear functional requirements that must be met by the application. The required functionalities include appointment scheduling, device manufacturer tracking, client contact information management, and reminder notifications.

Unit testing will be performed by developers to verify that individual components or segments of the code work as intended. Developers will write and execute unit tests for each individual function or method. For example, the appointment scheduling function will be tested to ensure it correctly handles dates, times, and availability, and the notification system will be tested to ensure reminder emails or SMS messages are triggered correctly. The developers will use frameworks such as JUnit or PyTest to test these individual functions before integrating them into the overall application. Once unit tests are complete, integration testing will be performed by the QA team to ensure that different modules of the application work together as expected. This is particularly important for interactions between the user interface (UI) and backend systems. The QA team will simulate end-to-end interactions, such as scheduling an appointment, modifying client details, and sending a notification to ensure these processes integrate seamlessly.

After integration testing, system testing will be conducted by both the QA team and external testers to verify that the entire application works as a complete system and meets the functional specifications outlined in the requirements. Full system tests will be run to validate all workflows across the app. This includes ensuring all features are accessible and operational from the user interface. The next stage is User Acceptance Testing (UAT), which will be conducted by Tech Gopher LLC.'s end-users, including technicians, administrative staff, and customer support teams. The product owner and project manager will also be involved in facilitating communication and addressing any concerns. During UAT, users will go through common tasks such as scheduling an appointment, updating client contact details, or rescheduling appointments, while confirming that the system behaves as expected. This stage will allow Tech Gopher LLC. to validate whether the scheduling application meets their operational needs in real-world scenarios.

Regression testing will be performed throughout the development and implementation process to ensure that new changes or fixes do not introduce errors into existing features. The QA team and developers will run regression tests after each update or bug fix to make sure that the system remains stable. For instance, after fixing bugs related to appointment scheduling, regression testing will ensure that the fix does not break other related functionality, such as the reminder notifications.

The timeline for testing will include unit testing during the development phase, integration testing once individual modules are completed, and system testing after integration. UAT will occur just before the full deployment, allowing Tech Gopher LLC.’s end-users to test the system under real-world conditions. Regression testing will be ongoing throughout the development process. As part of the Agile approach, feedback will be continuously solicited from both the testing team and Tech Gopher LLC.'s users during UAT. Any issues discovered will be documented, prioritized, and addressed in a timely manner to ensure the system meets the expectations of the customer before final deployment.

Once all testing phases are completed and any identified issues are resolved, Tech Gopher LLC. will perform the final validation during UAT. After their acceptance and sign-off, the application will be officially handed over to the customer for full deployment and use. The project manager will ensure that all necessary documentation, including user manuals and technical support information, is delivered.

Environments and Costs

Programming Environment Overview

To develop the TechGopher Scheduler application, a specific set of hardware and software environments will be required. The development environment will include tools such as IntelliJ IDEA for coding, Maven libraries for dependency management, and Java as the primary programming language. Version control through Git will be used to manage source code, while JUnit will be utilized for testing the application. For deployment and production, a cloud-based server or on-premise infrastructure will be necessary to host the application and database, that has security measures like authentication and encryption ready off the shelf. The application will be rolled out in phases to ensure a smooth transition and minimize risks.

Environment Costs Overview

Regarding the costs associated with the software application, there will be both startup and recurring expenses. Startup costs include licensing fees for development tools such as IntelliJ IDEA and any necessary Maven dependencies, as well as initial setup fees for cloud infrastructure and database storage. Additional expenses will include project management tools for tracking progress and collaboration, and security implementations such as SSL certificates. Recurring costs will involve cloud hosting fees, database maintenance, ongoing licensing or subscription costs for development tools and frameworks, security monitoring, and updates. Additionally, support and maintenance costs will be negotiated post-deployment to ensure the application continues to function effectively.

Human Resource Costs and Requirements

In terms of human resources, the project will require a project manager (PM), a designer, two software engineers, and one QA specialist. The PM will work part-time throughout all phases of the development lifecycle, with the lowest activity expected during the implementation phase. Their total activity is estimated to be 100 hours, which will cost $6,000 at an hourly rate of $60/hour. The designer's hourly rate is also $60/hour, but their activity is mostly confined to the system design phase and is estimated at 40 hours for a project of this size.

The two software engineers will account for the majority of the development costs. Their hourly rate is currently $70/hour, and the total length of the implementation phase is estimated to be three business weeks.

Finally, the services of the QA specialist will be required during the testing phase and should not exceed 40 hours. At an hourly rate of $40/hour, this would amount to $1,600.

Given these estimates, the total cost for human resources for this project is as follows:

Project Manager: $6,000 (100 hours × $60/hour)

Designer: $2,400 (40 hours × $60/hour)

Two Software Engineers: $28,800 (2 engineers × 3 weeks × 40 hours/week × $70/hour)

QA Specialist: $1,600 (40 hours × $40/hour)

Total Human Resource Cost: **$38,800**

To successfully complete the scheduling application for Tech Gopher LLC., a specific set of hardware and software environments will be required. This will include the development environment, as well as the deployment and production environments.

Hardware Costs and Requirements

Workstations: Developers and testers will need high-performance computers with at least 8GB of RAM, Intel i5 or equivalent processors, and 500GB SSD storage to support the development, testing, and debugging processes. For 5 developers and 2 testers, assuming the cost of each high-performance workstation is around $1,000, the total cost would be $7,000 for hardware.

Server Infrastructure: A cloud-based solution (e.g., Amazon Web Services (AWS), Google Cloud Platform (GCP), or Microsoft Azure) will be necessary to host the application’s backend, databases, and other services to support the Appointment Scheduling needs of the 12 TechGopher locations. At least one virtual private server (VPS) will be required for staging and production environments. The minimum specifications might include 4 vCPUs and 16GB of RAM to handle expected workloads.

A typical cloud hosting service (AWS, GCP, Azure) for a medium-sized application might cost between $50 to $500 per month, depending on the storage, compute, and bandwidth required. AWS

EC2 instances with 4GB of RAM and 20GB of storage typically cost about $50/month, while more robust configurations for scaling can increase these costs.

Total Hardware Costs: $8,000 initial

~$500 monthly

Software Requirements

The application will be developed using Java and FXML for the front-end user interface, and Java and MySQL for the backend. Maven will be employed for library management.

Database: MySQL or PostgreSQL for relational database management, with SQL Workstation for managing database interactions in the back-end.

Version Control: Git with repositories hosted on GitHub or GitLab to manage source code, handle collaboration, and ensure code versioning.

Testing Tools: JUnit or Mocha/Chai for automated unit testing, and Postman for API testing.

Deployment: Docker for containerization of the application, ensuring smooth deployment across different environments.

Support Contracts: For long-term operational stability, a support contract for server maintenance, security patches, and software updates might be required, typically costing $100 to $500 per month based on the scale of operations.

Total Software Costs: ~$500 Monthly

TechGopher Scheduler - A Client Scheduling Application

Total Environment Costs: ~$45,800 outlay

~$1000 monthly maintenance cost

Project Timeline

Phase	Milestone/Task	Deliverable	Description	Dates
Pre-development	Confirmation of requirements with customer	High level design documents and consensus on outline	Developers will review business needs outlined in requirements documentation and create high level design documents for the application.	3/14/2025-3/31/2025
System Design	Approved high-fidelity wireframes	High-fidelity wireframes	Developers will design and create high fidelity wireframe to demonstrate application flow	4/1/2025-4/7/2025
System Design	Confirmation of database requirements, high-fidelity wireframes	SQL Database Diagram	Developers will define and model entities that can support database interface and presentation in application	4/7/2025-4/11/2025
System Design	Design and outline unit test plan	Test plan	PM will work with developers to create functional unit and acceptance testing	4/11/2025-4/15/2025
Implementation	Distribution of Final Alpha Executables of Scheduling Application	Alpha Version of the Application is distributed to test users specified	Developers will implement feature requirements matching design documents and class diagrams to create application according to specs. Unit testing begins and runs without errors.	4/15/2025-5/7/2025

Testing Phase	Functional Testing	Functional Tests are Executed Successfully.	QA Engineer and developers runs test according to unit test specification	5/7/2025-5/14/2025
Testing Phase	Acceptance Testing	Approval documents from Customer signifying user acceptance	Customer will run acceptance testing to verify that the application meets requirements.	5/14/2025-5/20/2025
Deployment	Prepare deployment environment and deploy application	Deployed application ready to be used by customer	QA Engineer and Development team run system test on deployed hardware	5/20/2025-6/1/2025
Maintenance	Configure maintenance plan for future updates, improvements, and maintaining the application in the future	Maintenance service contract negotiation	Discuss and finalize terms of service for ongoing maintenance with the customer.	6/3/2025