**D424 – Software Capstone**

**Task 2**



|  |  |
| --- | --- |
| **Capstone Proposal Project Name:** | WGU Organizer |
| **Student Name:** | Blake Geard |

# 

Contents

[**Business Problem** 3](#_Toc183109713)

[**SDLC Methodology** 5](#_Toc183109714)

[**Deployment Plan and Outcomes** 8](#_Toc183109758)

[**Project Timeline** 10](#_Toc183109786)

[**Environments and Costs** 11](#_Toc183109787)

[**Validation and Verification** 12](#_Toc183109811)

[**Sources** 14](#_Toc183109812)

# **Business Problem**

**Business Problem**

Students in flexible, self-paced degree programs, like those offered at WGU, face significant challenges in staying organized, tracking academic progress, and managing deadlines effectively. These challenges are compounded by the lack of a centralized, user-friendly solution that integrates key features such as task management, progress tracking, and automated reminders.

**The Customer**

The target audience for this application includes students aged 25–40, many of whom are working professionals or managing family responsibilities alongside their education. These individuals need a tool that complements their busy lifestyles, streamlining the management of academic tasks to ensure deadlines are met and progress is visualized effectively. The customer base includes WGU's diverse student population, which operates within a flexible, competency-based educational model.

**Business Case**

Current solutions often lack the integration of core features needed by students in flexible academic environments. Existing systems may provide progress tracking or reminders but fail to deliver a comprehensive, unified experience tailored to the needs of WGU students.

The proposed application will act as a centralized mobile platform that addresses these pain points. It will:

1. Track term progress, course milestones, and assessments.
2. Send reminders for deadlines to reduce missed submissions.
3. Offer an intuitive interface designed for non-technical users, ensuring ease of adoption.

**Fulfillment**

The application will feature:

* Term and Course Management: Students can input their academic terms, courses, and corresponding deadlines.
* Progress Tracking Dashboard: A visual representation of completed and pending tasks to help students monitor their progress.
* Automated Notifications: Alerts and reminders for upcoming deadlines.
* Cross-Platform Access: A seamless experience across Android and iOS devices.

The app will integrate securely with external academic systems (if required in the future) and will employ SQLite for offline data storage, ensuring reliable access to information. The ultimate goal is to improve productivity, reduce organizational overhead, and support students in achieving academic success.

# **SDLC Methodology**

# The *Agile* methodology will be used for this project because of its iterative nature, which allows for continuous feedback and adaptability. Agile is well-suited for projects that require a user-centric approach and enables early delivery of core functionality while allowing for future enhancements.

# **How Agile Will Be Applied:**

# Iterations (Sprints): Work will be broken into short, manageable sprints, each delivering a functional aspect of the app.

# Frequent Feedback: Feedback from a small group of WGU students will guide iterative improvements.

# Incremental Delivery: Key features (e.g., progress tracking, notifications) will be developed and tested incrementally to ensure quality.

# **Planning Phase**

# **Planning Phase**

# **Deliverables:**

# Project Requirements Document: A detailed document outlining the functional and non-functional requirements of the application, gathered from initial user stories and interviews. This ensures alignment with customer needs and serves as a reference for all development phases.

# Initial Wireframes: Rough, low-fidelity representations of the app’s structure, showing the primary navigation and layout. These provide a visual reference for the development team and stakeholders.

# **Description:**

# The Project Requirements Document defines the project scope and user expectations, ensuring developers understand the objectives. The wireframes act as a blueprint for designing the app's interface, enabling early feedback from stakeholders.

# **Design Phase**

# **Deliverables:**

# High-Fidelity Mockups: Polished UI/UX designs that reflect the app's visual style, color scheme, and layout. These mockups will help stakeholders visualize the final product.

# Database Schema: A detailed structure for the SQLite database, including tables for terms, courses, deadlines, and user preferences.

# API Design Documentation: Specifications for internal and external interactions between app components, ensuring efficient and consistent data flow.

# **Description:**

# This phase finalizes the visual and functional design of the app. Mockups ensure the app meets user expectations, while the database schema and API design provide the technical foundation for development.

# **Development Phase**

# **Deliverables:**

# MVP (Minimum Viable Product): A functional version of the app with core features, including term and course management, notifications, and progress tracking.

# Sprint Reports: Summaries of completed work and progress, providing transparency and accountability.

# **Description:**

# The MVP focuses on delivering essential functionality to students, allowing early feedback and usability testing. Sprint reports ensure stakeholders remain informed about project progress.

# **Testing Phase**

# **Deliverables:**

# Test Cases: A comprehensive list of scenarios for unit, integration, and user acceptance testing (UAT).

# Bug Reports: Documents detailing identified bugs, their severity, and the steps to reproduce them.

# **Description:**

# Test cases ensure that all functionalities are thoroughly evaluated, while bug reports allow developers to prioritize fixes. User acceptance testing will involve a small group of WGU students to validate usability and effectiveness.

# **Deployment Phase**

# **Deliverables:**

# Final Software Release Package: The completed application, packaged for distribution on the Google Play Store and Apple App Store.

# User Documentation: Step-by-step guides and FAQs to help students install and use the app.

# Description:

# The final software package ensures students can access and install the app seamlessly. User documentation simplifies onboarding and reduces the need for technical support.

# **Maintenance Phase**

# **Deliverables:**

# Update Logs: Records of periodic updates, feature enhancements, and bug fixes.

# Usage Analytics Reports: Data insights on app usage to inform future improvements.

# **Description:**

# This phase focuses on long-term support for the app. Update logs demonstrate ongoing development efforts, while analytics reports identify areas for optimization

# **Deployment Plan and Outcomes**

# **Deployment Plan**

# The deployment of the proposed mobile application will follow a structured process to ensure a seamless transition into production. The steps below outline how the application will be prepared, validated, and released to end users:

# Staging Environment Testing:

# Before deployment, the application will undergo final testing in a staging environment that mirrors the production environment. This ensures that the app functions correctly under production-like conditions. Test cases developed during the Testing Phase will be executed here to validate features, resolve any remaining bugs, and ensure system stability.

# Validation and Verification:

# Validation: User Acceptance Testing (UAT) will involve a group of WGU students who will test the application to ensure it meets the requirements and provides a satisfactory user experience. Feedback from UAT will be used to make final adjustments.

# Verification: Final functionality, performance, and security tests will be conducted to ensure compliance with the original requirements and to validate data integrity and responsiveness.

# Deployment to Production:

# The app will be deployed to the Google Play Store and Apple App Store using standard submission processes for each platform. These include submitting the app for review by the respective stores, adhering to their guidelines, and ensuring compliance with data privacy regulations. The deployment will occur in the following stages:

# Initial Deployment: Limited release to a small group of pilot users to monitor performance and collect early feedback.

# Full Deployment: Full-scale release to all WGU students after successful completion of the pilot phase.

# Communication with End Users:

# An announcement email and in-app tutorial will guide students on how to install and use the application. The communication will include links to user documentation, FAQs, and support contacts for troubleshooting.

# **Roles and Responsibilities**

# The deployment process will involve multiple roles to ensure success:

# Developer (Blake Geard): Responsible for building and packaging the app for deployment and submitting it to app stores. Also ensures bug fixes and final optimizations are completed.

# Quality Assurance Tester: Responsible for staging environment testing and verification of app functionality and performance.

# Customer Support Representative (Fictional Role): Handles user inquiries, feedback, and troubleshooting during the initial rollout phase.

# **Dependencies and Timing**

# Pre-Deployment: Deployment can only occur after the successful completion of the Testing Phase, including staging environment testing and UAT feedback incorporation.

# Pilot Phase: The pilot deployment will depend on timely app store approvals and user feedback from pilot users.

# Full Deployment: The full release is dependent on the resolution of any pilot phase issues.

# **Anticipated Outcomes**

# Enhanced Productivity for Students: Students will be able to manage their academic schedules, deadlines, and progress seamlessly, reducing missed deadlines and boosting success rates.

# High Adoption Rate: The app’s intuitive design and WGU-specific customization are expected to drive significant adoption among students.

# Positive Feedback: Early feedback from the pilot deployment will validate the app's usefulness and provide actionable insights for further enhancements.

# Long-Term Improvement: Usage analytics gathered post-launch will inform updates and optimizations to maintain app relevance and usability.

# **Project Timeline**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Phase | Milestone/Task | Deliverable | Description | Dates |
| Pre-development | Task 1: Requirements Gathering | Project Requirements Document, Initial Wireframes | Document user stories and create low-fidelity wireframes to establish app scope and layout. Dependency: None. Resource: Blake Geard (Developer/Designer). | 12/1/2024 – 12/3/2024 |
| Design | Task 2: Design Finalization | High-Fidelity Mockups, Database Schema, API Design | Develop polished UI designs, define database structure, and document API interactions. Dependency: Completion of Task 1. Resource: Blake Geard (UI/UX Designer, Database Planner). | 12/4/2024 – 12/8/2024 |
| Development | Task 3: Core Feature Development | MVP (Minimum Viable Product), Sprint Reports | Implement key features like term/course management and notifications. Dependency: Completion of Task 2. Resource: Blake Geard (Full-Stack Developer). | 12/9/2024 – 12/19/2024 |
| Testing | Task 4: Testing and Debugging | Test Cases, Bug Reports | Execute unit, integration, and user acceptance testing. Document bugs and fixes. Dependency: Completion of Task 3. Resource: Blake Geard (Tester). | 12/20/2024 – 12/23/2024 |
| Depoyment | Task 5: App Deployment | Final Software Release Package, User Documentation | Publish app on Google Play and Apple App Stores, and provide user onboarding guides. Dependency: Completion of Task 4. Resource: Blake Geard (Developer). | 12/24/2024 – 12/25/2024 |
| Maintenance | Task 6: Post-Launch Support | Update Logs, Usage Analytics Reports | Monitor app performance, release updates, and analyze usage data for ongoing improvements. Dependency: Completion of Task 5. Resource: Blake Geard (Developer/Support Specialist). | Begins 12/26/2024 (Ongoing) |

# **Environments and Costs**

## **Programming Environment**

## The development and deployment of the proposed application will require the following hardware and software environments:

## Development Environment:

## Framework: .NET MAUI, a cross-platform framework enabling development for Android and iOS from a single codebase.

## Database: SQLite, a lightweight and efficient database for offline data storage.

## Integrated Development Environment (IDE): Visual Studio Community Edition, a free, feature-rich tool for coding, debugging, and project management.

## Version Control: GitHub for source control and collaborative code management.

## Test Environment: Simulated Android and iOS environments provided by Visual Studio for debugging and testing.

## Production Environment:

## Target Platforms: Google Play Store (Android) and Apple App Store (iOS).

## Hardware Requirements: The app will run on modern Android and iOS smartphones (minimum requirements: Android 8.0 or iOS 12).

## **Environment Costs**

## The development and production costs are minimal due to the use of free tools and platforms wherever possible. The estimated costs include:

## One-Time Costs:

## Google Play Developer Fee: $25 (one-time).

## Apple Developer Program Fee: $99/year for app deployment.

## Hardware for Testing: A mid-range Android device ($200) and an iOS device ($400) will be purchased to test the application on physical devices.

## Ongoing Costs:

## Cloud Database Hosting: While SQLite is used for local data storage, a future cloud integration may incur hosting fees (~$100/year). This cost is deferred for initial development.

## Total Estimated Costs:

## Initial Development: $724 (Google and Apple fees + testing devices).

## Annual Maintenance: $99 (Apple Developer Program fee).

## **Human Resource Requirements**

The project will be developed entirely by Blake Geard, serving as the sole developer and project manager. Labor costs are estimated based on the time required to complete the project:

1. Estimated Hours:

* Planning and Design: 14 hours.
* Development: 40 hours.
* Testing and Debugging: 10 hours.
* Deployment: 4 hours.
* Documentation: 4 hours.
* Total Development Time: 72 hours.

1. Cost of Labor:

* As the project is being developed independently, no external labor costs are incurred. However, for budgeting purposes, if this were a professional project, labor could be valued at $50/hour, resulting in an estimated labor cost of $3,600 for development.

# **Validation and Verification**

**Methods for Validation and Verification**

To ensure the proposed software product meets customer needs and performs as intended, the following validation and verification methods will be employed:

**Unit Testing:**

* Purpose: Verify that individual components (e.g., term/course management, notification triggers) function as expected.
* Method: Each function or module will be tested in isolation using automated test scripts and manual checks. For example, adding, editing, and deleting courses will be validated independently.
* Performed By: Developer (Blake Geard).

**Integration Testing**:

* Purpose: Ensure that all app components work seamlessly together, such as interactions between the SQLite database, notification system, and UI.
* Method: Simulate real-world use cases where multiple features interact (e.g., a student updates a course, which triggers a notification).
* Performed By: Developer.

**System Testing:**

* Purpose: Test the entire application in an environment that mimics production to validate performance, security, and overall functionality.
* Method: Use both simulated and real devices to ensure consistent performance across Android and iOS platforms. Testing will cover edge cases (e.g., invalid input, unexpected usage).
* Performed By: Developer and Quality Assurance Tester.

**User Acceptance Testing (UAT):**

* Purpose: Validate that the application meets customer needs and provides a satisfactory user experience.
* Method: A small group of WGU students will use the app in a controlled environment and provide feedback on usability, design, and functionality. Issues will be documented, prioritized, and addressed before final deployment.
* Performed By: WGU student testers and developer for analysis of feedback.

**Justification:**

These testing methods ensure thorough validation and verification by addressing every level of the application:

1. Validation: Ensures the application meets functional requirements and provides a positive user experience.
2. Verification: Confirms technical accuracy and that all components work as intended without bugs or performance issues.

**Analysis of Test Results**

Once testing is complete, the following steps will be taken to analyze the results and ensure that all issues are resolved effectively:

1. Documentation of Test Results:

Each test case will include the following:

* Input data and expected outcomes.
  + Actual outcomes (pass/fail).
  + Any deviations from expected results.
* Results will be logged in GitHub Issues for tracking, with each bug assigned a priority level:
  + High: Critical errors (e.g., crashes, data loss).
  + Medium: Functional but incorrect behavior (e.g., wrong notifications).
  + Low: Minor UI inconsistencies or non-disruptive errors.

1. Bug Resolution Process:

* Bugs will be addressed in priority order during subsequent development sprints.
* Once resolved, the fixes will be re-tested in a staging environment to ensure no new issues were introduced.

1. Acceptance Criteria:

* The application will be considered ready for deployment when all high-priority bugs are resolved, and the app meets predefined functional and performance benchmarks, including:
* Accurate tracking of courses, terms, and deadlines.
* Notifications triggering on schedule.
* Stable performance across Android and iOS devices.

1. Final Report:

* A comprehensive report will summarize:
* Total test cases executed.
* Bugs identified, fixed, and remaining (if any).
* UAT feedback and implemented changes.
* Confirmation that the application meets all requirements.

**Sources**

No sources were used for this report.