Implementation Plan for Project: Indy Student Life

Green Team

Team Members: Emmanuel Akinseye, Ann Chen, Ryan Engelken, Maurice Farr, and Harjot Singh

Department of IT, Ivy Tech Community College

Dr. Carver

SDEV-265

April 28, 2025

Tables of Contents:

Introduction

This implementation plan details the organized strategy used to create Indy Student Life, a web-based tool aimed at assisting university and college students in Indianapolis, IN, in finding affordable food options within a 4000-meter (2.5-mile) radius of their campus and locating suitable on-campus study spots. The document outlines the software's development process, the strategies employed, and provides guidance for its deployment. It serves as a resource for both the development team and stakeholders or users interested in understanding or installing the software.

The document covers several key areas, including:

- **Implementation Methods Pros and Cons** – Comparison of real-world implementation methods and justification of the selected one
- **Details:** Description of how the software requirements were implemented
- **Installation Instructions:** Steps to install and run the application
- **Summary:** Recap of the document's key points

This information is intended to clarify the reasoning behind important implementation choices, review how each requirement was met, and facilitate the successful deployment of the software.

Implementation Methods

Several implementation strategies are commonly employed in software development projects. The main methods considered for this project include:

- **Waterfall Methodology**

  Pros: It is beneficial for managing small projects due to its structured approach and distinct review processes. It features clearly defined deadlines for each development stage, making it easier to track progress. (GeeksforGeeks, 2024).

Cons: It is not suitable for projects that require ongoing modifications, as it lacks the flexibility to adapt to changes during the development process. (GeeksforGeeks, 2024).

- **Agile Methodology**

  Pros: It is known for its iterative approach, which enhances the quality of the final product by allowing for ongoing creative improvements throughout the development process. It is also characterized by its flexibility, as it does not heavily depend on initial documentation (GeeksforGeeks, 2024).

  Cons: One of its drawbacks is the absence of strict deadlines, which can lead to a lack of clarity and a defined project vision (GeeksforGeeks, 2024).

- **Lean Methodology**

  Pros: It is effective as it is cost-friendly, which helps teams accelerate the software development process and complete more projects in a shorter time frame (GeeksforGeeks, 2024).

  Cons: A notable drawback is the insufficient documentation related to business objectives, which can hinder clarity and alignment (GeeksforGeeks, 2024).

Chosen Strategy: Agile Methodology

Our team opted for the Agile Methodology due to its flexible and collaborative nature, which aligned well with the project's evolving scope and timeline. We employed tools such as Trello for task management, GitHub for version control, and Discord for communication, allowing us to adapt swiftly, assign tasks efficiently, and iterate effectively. While we considered the Lean method for its focus on essential features and minimizing extras, Agile proved to be a better fit for coordinating our efforts in an academic setting. Nonetheless, we still incorporated Lean principles by prioritizing quick value delivery and reducing unnecessary features.

Details

- **Requirement 1: Interactive Map Display**

  Leaflet.js was utilized to create an interactive map interface. The base maps are powered by data from OpenStreetMap contributors, adhering to the Open Database License (ODbL) license. To enhance visual distinction, custom icon sets representing campuses, restaurants, and study spots were incorporated. The map interface is designed to dynamically load markers based on JavaScript Object Notation (JSON) or database inputs, adjusting the view according to the campus selected by the user.

- **Requirement 2: Location Radius Filtering**

  Location data was verified against a central campus coordinate using a distance calculation in JavaScript, ensuring that only entries within 4,000 meters were included in the final dataset. Raw location data was stored and filtered using SQLite3 before being displayed on the front end.

- **Requirement 3: Categorized Listings (Food vs. Study Spots)**

  We developed a category-based filtering system using JavaScript functions that allow users to add or remove specific types of markers based on their preferences, such as displaying only restaurants, study spots, or both. This approach ensures a streamlined and personalized user experience, preventing information overload by presenting only relevant data at any given time.

- **Requirement 4: Desktop Compatibility and Responsive Design**

  To ensure compatibility with modern browsers and devices, the project utilized HTML5, CSS3, and JavaScript. The layout is designed to be responsive, employing flexbox and media queries to accommodate both mobile and desktop views. Special care was taken to maintain the accessibility of map controls and category toggles on smaller screens. Throughout the phased implementation, all components were rigorously tested across various browser types and sizes.

- **Requirement 5: Database Integration**

  We utilized SQLite3 to manage vendor information, including name, location coordinates, price category, and description. The database was managed through DB Browser and integrated into our development pipeline to ensure consistency. Although the website is static and hosted on GitHub Pages, we incorporated a SQLite3 database for data storage during the design and testing phases. This setup facilitated organized data storage for testing retrieval and mapping functionalities. While the live site does not connect to SQLite dynamically, the database structure is ready for future backend development.

- **Requirement 6: Collaboration and Version Control**

  The team utilized GitHub for managing version control and collaborative coding efforts. All project files are stored in a public repository, allowing for regular commits and effective branching strategies. This setup enabled team members to work on various modules at the same time, significantly reducing the likelihood of merge conflicts.

Installation Instructions

1. Open your terminal or command prompt and run:

```
git clone https://github.com/harjot3/IndyStudentLife.git
```

2. Navigate to Project Directory:

```
cd IndyStudentLife
```

3. Open Project in Visual Studio Code (or your preferred editor):

```
code .
```

4. Open index.html in a Browser:

Navigate to the main HTML file and open it. The interactive map should load, displaying food options within the defined radius.

5. (Optional) Set Up SQLite3 Database:

- Download DB Browser for SQLite.

- Open the provided .sqlite file (if available).

6. Link to Live Website:

https://harjot3.github.io/IndyStudentLife/goal.html

Summary

This document provides a comprehensive overview of the implementation of the Indy Student Life project. We discussed the project's goals, examined various practical strategies for execution, and justified our decision to adopt a phased approach. The project leverages open-source technologies such as Leaflet.js and OpenStreetMap, along with well-organized HTML, CSS, JavaScript, and SQLite for data storage. Additionally, we included straightforward installation instructions to help users set up the application.

By employing a modular, phased strategy and utilizing open-source resources, our team has successfully created a scalable and user-friendly platform for discovering food and study locations tailored for students in the Indianapolis area.

Work Cited

GeeksforGeeks. (2024, January 3). *What are software development methodologies: 15 key methodologies*. https://www.geeksforgeeks.org/what-is-software-development-methodology-15-key-methodologies/