GINA CODY SCHOOL OF ENGINEERING AND COMPUTER SCIENCE

SOEN 341 Software Process

Team Project, Winter 2024

Project Title: A Car-rental Web Application

Objective.

This project will help you to get a taste of software project management skills firsthand. You will follow the Agile development approach; take advantage of GitHub distributed version control plus access control, bug tracking, software feature requests, task management, continuous integration, and wikis to support your project management process. The project is divided into 4 incremental deliveries which we refer to as sprints based on Agile Scrum methodology, which will be used in this course.

The duration of the project is around 10 weeks; the development process is an adapted Agile with 3 to 4 weeks long iterations, 4 iterations in total. The first 2 weeks of the first sprint are for training and setting up your development environment.

Because of the short span of this project, you are not expected to deliver a marketable product, but the result should be at least a compelling middle-fidelity prototype that could serve as the basis for building a real product. Check these two links on prototype fidelity quite helpful: A Guide to Prototype Fidelity:

- https://www.webfx.com/blog/web-design/design-mockup-fidelity/
- https://www.webfx.com/blog/web-design/wireframes-vs-prototypes-difference/

Description.

A car rental application is a software platform designed to facilitate the process of renting vehicles for short periods, typically ranging from a few hours to a few weeks. The application serves as an interface between customers looking to rent vehicles and the car rental company offering those services.

We three identify primary users: Customers, Customer service representatives (CSR), and system administrator(s).

List of main use cases organized by user type.

1. Customer
   - 1.1. Browse vehicles for rent: A catalog of rental vehicles
   - 1.2. Start a reservation: After providing a location (postal code, city, or airport) and a pickup and return date, the customer will be shown vehicles that match the specific criteria such as type (CARs, SUVs, Vans, Trucks) category (compact, standard, intermediate, etc.), and price range.
     - 1.2.1. During the reservation, a customer can add extra equipment at an additional price
   - 1.3. View/Modify/Cancel reservation
   - 1.4. Find a branch: The customer provides a postal code or Airport to find the nearest branch
   - 1.5. Rating and review: Customers can provide feedback and ratings for the rented vehicles and overall rental experience.
2. Customer service representative
   - 2.1. Check-in process for customers with or without reservation
     - 2.1.1. if the customer did not make a reservation, he creates a new reservation in the system
     - 2.1.2. if the customer has made a reservation, he confirms the reservation in the system by verifying the customer's reservation, and customer identification; then, it proceeds to the rental agreement review and payment processing.
   - 2.2. Check-out process. After physically inspecting the vehicle, the CSR reviews the rental agreement ensuring that the terms and conditions of the rental are met; processes the final billing based on the rental duration, additional services, and any applicable fee; performs the payment settlement and confirms the completion of the return process in the system.
3. System administrator
   - 3.1. CRUD operations on vehicles
   - 3.2. CRUD operations on user accounts
   - 3.3. CRUD operations on reservations

This is not an exhaustive list of features/users. Other kinds of users and features could be considered. Your project grade criteria consider originality and innovation. The highest marks will be given to teams who are thinking out of the box and include other relevant users and functionalities for eventual implementation when properly justified.

For the first sprint, you have to consider the following core features:

- **Start a reservation**
- **View/Modify/Cancel a reservation**
- **Browse vehicles for rent**

- **CRUD operations on users**
- **CRUD operations on vehicles**
- **CRUD operations on reservations**

You must have regular meetings with your team and post their "minutes" in the corresponding subfolder in your repository.
At the end of each sprint, each team member will submit a detailed log of their activities, which will be considered to evaluate the individual contribution.

Sprint 1 delivery instructions.

Create a **private repository** in GitHub named <team_name-soen341projectW2024> with all team's **document** deliverables in this repository (organized by subfolders: Sprint 1, Sprint 2, Sprint 3, and Sprint 4). This repository will contain all documents related to your project such as README file, project reports, meeting minutes, user stories, etc..
Add your TA as well as your instructor to the repository.

You must submit to Moodle, as a team, one page including your team name, team members with IDs, and a link to your repository. The rest of the work is stored and evaluated directly in your GitHub repository.

The table below provides details on the activities and the corresponding deliverables that must be present in your repository. A detailed grading rubric can be accessed here. This will be used by the markers.

Activities	Sprint Details	Weight

1. GitHub setup and initialization 2. README file 3. Project approach and technology 4. User stories and Task Breakdown 5. Plan for the next Sprint	☐ README file with ○ Description of the Project ○ Team Members and Roles ○ Project Approach and Technology ☐ Project approach and technology (Use template) ☐ 6 user stories backlog for Sprint 1 (Use GitHub issues) ☐ Task Breakdown (derived from user stories, and assigned to a team member each of them). (Use GitHub issues) ☐ Detailed log of each team member's contribution including time spent on each activity (document) ☐ Meetings Minutes file (minutes files should be named <teamName_Sprint#_meetingnumber_meeting_date> ☐ Plan for the next Sprint on a wiki page	3%
	All members of the team should contribute equally to the project and all contributions must be traceable on GitHub. For informal communication among the team members, you use the private forum in Moodle designed for this purpose.	

Appendix A. Project Approach and Technology Stack Selection Template

1. Project Overview

1.1 Project Objectives

- Clearly define the goals and objectives of the project.

1.2 Scope

- Outline the scope of the project, including key features and functionalities.

1.3 Target Audience

- Identify the intended users and their needs.

2. Project Approach

2.1 Development Methodology

- Choose a development methodology (e.g., Agile, Waterfall) and justify the selection based on project requirements.

2.2 Project Timeline

- Create a high-level timeline outlining major milestones and deadlines.

2.3 Collaboration and Communication

- Define communication channels and collaboration tools for the project team.

3. Technology Stack

3.1 Backend Frameworks

3.1.1 Framework A

- Description: Brief overview of Framework A.
- Rationale:
  - Justification for choosing Framework A.
  - Consider factors such as community support, scalability, and ease of integration.
- Qualitative Assessment:
  - Strengths
  - Weaknesses
  - Use Cases

3.1.2 Framework B

- Description: Brief overview of Framework B.
- Rationale:
  - Justification for choosing Framework B.
  - Consider factors such as performance, security, and maintenance.
- Qualitative Assessment:
  - Strengths
  - Weaknesses
  - Use Cases

3.1.3 Framework C

- Description: Brief overview of Framework C.
- Rationale:
  - Justification for choosing Framework C.
  - Consider factors such as community support, learning curve, and extensibility.
- Qualitative Assessment:
  - Strengths
  - Weaknesses
  - Use Cases

3.2 Frontend Frameworks

3.2.1 Framework X

- Description: Brief overview of Framework X.
- Rationale:
  - Justification for choosing Framework X.
  - Consider factors such as user interface capabilities, responsiveness, and cross-browser compatibility.
- Qualitative Assessment:
  - Strengths
  - Weaknesses
  - Use Cases

3.2.2 Framework Y

- Description: Brief overview of Framework Y.
- Rationale:
  - Justification for choosing Framework Y.
  - Consider factors such as modularity, performance, and community support.
- Qualitative Assessment:
  - Strengths
  - Weaknesses

- Use Cases

3.2.3 Framework Z

- Description: Brief overview of Framework Z.

- Rationale:

  - Justification for choosing Framework Z.

  - Consider factors such as ease of integration, component libraries, and developer experience.

- Qualitative Assessment:

  - Strengths

  - Weaknesses

  - Use Cases

4. Integration and Interoperability

4.1 Backend-Frontend Integration

- Outline the strategy for integrating the chosen backend and frontend technologies.

4.2 Third-Party Services

- Identify any third-party services or APIs that will be integrated into the project.

5. Security Considerations

- Provide an overview of security measures and considerations for both the backend and frontend.

6. Conclusion

- Summarize the chosen project approach and technology stack, highlighting key reasons for the selections.

This template provides a structured framework for selecting a project approach and technology stack, including a qualitative assessment of backend and frontend frameworks. Customize the template according to the specific needs and details of your project.

Appendix B. Plan for the next sprint details.

Here are some detailed instructions about planning for Sprint 2.

Now that you have written the 6 user stories and broken them into tasks, you must allocate story points to the user stories (estimate effort).

For the planning of the next sprint, you will create a table with the following columns:

- Issue # from your repository.
- User story/Task title.
- Story points associated with a User story (US)/task.
- Due date.
- Associated tasks/Task description.
- Priority (low, medium, high) based on your discussion with your TA.
- Risk (low, medium, high) and a brief explanation.
- Responsible. The team member who is accountable for the task.

As an example, consider the project from a previous term.

https://docs.google.com/spreadsheets/d/1XJVPkzoddaY0HJMaGUNn_-xZByVyE6RBzXC0nnBDX_g/edit?usp=sharing

Remember that this table is the result of your team meetings, and the discussions with the TA, and/or instructor. All decisions must be documented in the minutes of your meetings, for example, how do you estimate story points for each task.