DEVELOPMENT PLAN

Dining Services App

UNIEATS BY DIVPLUSPLUS

**PREPARED BY DIVPLUSPLUS TEAM:**

YONATAN AZARAF – 2460755

ROHAN CHHIKA – 2543404

NAFTALI DINER – 2144598

DEVESH CHIBA – 2537948

SHAYUR GOVIN -2558583

SAYJ RAMSAMY - 2551879

**Project Name:** Dining Services App

**Prepared By:** *divplusplus* Team

**Project Timeline:** 6 August 2024 – 8 October 2024

**Detailed Project Overview**

The student will be welcomed to our web application with an about us page with the option to signup/login. Once the student has selected signup/login, the student will then be redirected to our 3rd party authentication whereby they will be required to enter an email and password to login. Signing up/logging in is available through google too. If the student is not a user within our system, the student will then be required to sign up and create an account. Once the student creates an account, a verification link will be sent to them to verify their email address. Additionally, to logging in, a “forgot password” option is available and can be used if needed. Upon a successful login, the student will be redirected to our home page where they can see a list of all the restaurants available. The student can click the “View my credits” button, where they can see their available credits and their purchase history. The students can search for their desired restaurant, or they can scroll through the list of restaurants and pick a restaurant. Once the student selects their restaurant of choice, the student will be redirected to a page where they can see the menu of that restaurant, along with reviews of that restaurant. The student can filter the menu by their dietary requirements/preferences. Each item on the menu will be listed with the price, description and ingredients. The students will also have the option to make a reservation for the selected restaurant whereby they will be asked to fill in information regarding the number of guests and the date and time of their booking. The student can also leave a review on the restaurant where they are redirected to a page where they can leave a review and a star rating. Lastly the student can view their cart which will have all the items they selected from the restaurant with the total price, and they will have the option to check-out, whereby they will be redirected to a page that will inform them of their estimated time of collection, their order number and their available balance of credits after the purchase. The student is then able to log out of their account.

**Development Environment**

GitHub will be used as our development environment's repository host, and Microsoft Azure will be used to deploy our apps. We will use the React library for front-end programming and Express and NodeJS for back-end development. MongoDB will be used to host our databases. Finally, to receive data from other teams that will benefit our application, we will be integrating with them. Other teams will be able to integrate our application with theirs.

**API Specification**

The backend API for the UNIEATS website is developed using Node.js and Express.js, providing a robust framework for building a RESTful API that seamlessly interacts with the React frontend and MongoDB database. Hosted on Azure Web Apps, this API ensures high availability and scalability, allowing for efficient handling of requests from the client-side application. The API exposes several endpoints for essential functionalities, including user authentication, menu retrieval, order placement, and feedback submission. Each endpoint follows RESTful conventions, utilizing standard HTTP methods such as GET, POST, PUT, and DELETE to manage resources effectively. The integration with MongoDB facilitates a flexible and schema-less database structure, enabling dynamic content management and efficient data retrieval. To enhance security, JSON Web Tokens (JWT) are employed for user authentication, ensuring that sensitive data is protected during transmission. The API is designed with performance and maintainability in mind, featuring comprehensive logging and error handling mechanisms, allowing for easy debugging and monitoring of application health. This architecture provides a solid foundation for delivering a responsive and user-friendly dining services platform for the campus community.

**User Stories:**

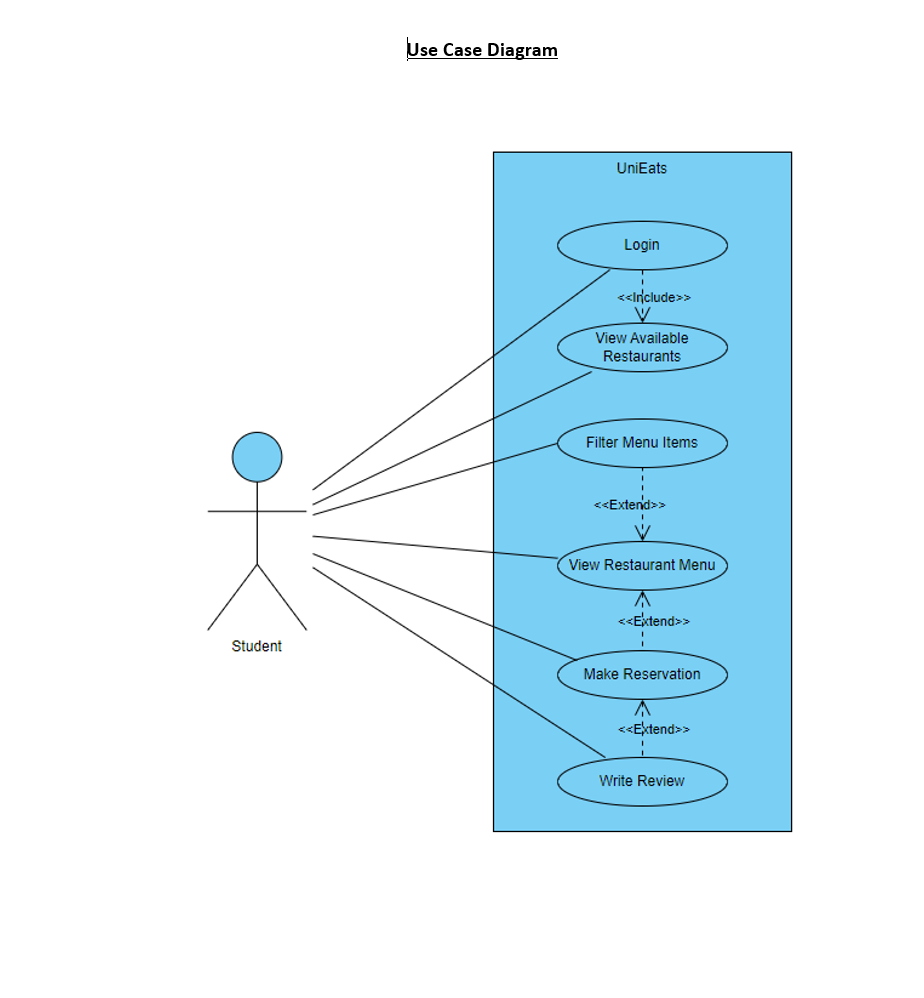
1. As a student, when I login to my profile, I will be able to view a list of all the restaurants available.

2. As a student, when I select my desired restaurant, I can see the menu.

3. As a student, I can filter the items on the menu, so that I am able to view the items that fit my dietary requirements.

4. As a student, I can make a reservation at my selected restaurant so that I can secure my spot at the restaurant.

5. As a student, I can write a review on the restaurant so that the restaurant can use my feedback.



**Possible Integrations**

In developing our campus app's dining hall services, we see exciting opportunities for integration with other key services. We plan to collaborate with the transportation team to incorporate bus schedules, ensuring students can easily coordinate their travel with dining times. Additionally, we aim to integrate with the events team to facilitate catering services, making it seamless for students and organizers to manage dining arrangements for campus events. These integrations will enhance the overall user experience, providing a more connected and efficient service platform for the campus community.

**Key Feature Milestones**

1. **Project Initiation (6 August 2024)**

o **Gather all team members:** Outline project goals and scope. Establish communication methods.

o **Project Scope:** Clearly outline the desired outcomes of the project, and what will be included and excluded.

1. **Requirements Gathering and Analysis (12 August 2024)**

o **Meeting Stakeholders:** Engage with stakeholders to gather detailed requirements and expectations, taking note of their requirements and any constraints they might have.

* + - **Create a Product Backlog:** Compile user stories into a prioritized list that will be used as a guideline for our project development.

1. **Design Phase (13 August 2024 – 20 August 2024)**

o **Create Architecture Diagram:** Create a visual representation of the system’s architecture which includes the main components, the component interactions, and data flow.

o **Design Database Schema:** Outline the structure of the database.

1. **Development Phase – Sprint 1 (21 August 2024 – 30 August 2024)**

* **Implement Login and Signup Page for students:** Develop and test pages that allow applicants to create accounts and log in to the system. This should be through a 3rd party authentication such as Auth0. Securing communication between frontend and backend using JSON web tokens.
* **Implement landing page after login/signup**: Create a landing (home) page where the student can see a list of available restaurants.

1. **Development Phase – Sprint 2 (31 August 2024 – 7 September 2024)**
   * + **Implementation of menus for each restaurant**: Create menus for each restaurant.
     + **Describe each menu item**: Each item will have a description including the price and ingredients of the item.
2. **Development Phase – Sprint 3 (8 September 2024 – 12 September 2024)**
   * + **Add Filter option:** implement a filter option so students can filter the menu according to their dietary requirements.
     + **Conduct Integration Testing:** Test the interaction between different components of the system. This is to verify that they work together as expected.
3. **Development Phase – Sprint 4 (12 September 2024 – 17 September 2024)**
   * + **Implement a reservation feature:** create a booking feature so that students can reserve a table at their selected restaurant.
     + **Notify students before their reservation (Stretch goal):** develop a booking feature so that students are notified an hour before the reservation, reminding them of their booking.
     + **Add review option:** create a star rating system where students can leave a rating on their experience at the restaurant. Students can also leave a written review.
     + **Conduct System Testing:** Perform comprehensive testing of the entire system to ensure it meets all requirements and functions correctly.
4. **Testing and Quality Assurance - Sprint 5 (17 September 2024 – 1 October 2024)**
   * **Conduct User Acceptance Testing:** Involve users in testing the system in an actual setting to make sure it satisfies their requirements and expectations.
   * **Fix any bugs:** Resolve any issues found during testing to guarantee a dependable and seamless user experience.
   * **Complete Testing and Validation:** Complete any outstanding testing assignments and verify that the system is ready for implementation**.**
5. **Final Review and Deployment (Continuous Deployment using GitHub actions)**
   * **Complete the project review and handover:** Confer with stakeholders, confirm that all deliverables have been met, and turn over the system for usage in operations.

**Roles and Responsibilities (Please note that we are all full stack developers but also have our own responsibilities to manage.)**

* + Project Manager and Developer (Sayj Ramsamy)
  + Lead and Backend Developer (Yonatan Azaraf)
  + Frontend Developer (Rohan Chhika)
  + Backend Developer (Naftali Diner)
  + UI Designer (Shayur Govin)
  + Frontend Developer (Devesh Chiba)

In our project, although all members are full stack developers with the ability to work on both the frontend and backend of our web application, we’ve decided to assign specific responsibilities to ensure smooth progress and accountability. Each of us will take on a particular role, such as Project Manager, Frontend Lead, Backend Lead, Testing, UI/UX Lead, and Quality Assurance Lead. These roles will help streamline our workflow, as each member will focus on ensuring that their respective area is completed efficiently and effectively. **Despite these assigned roles, we all remain versatile, contributing to any part of the stack as needed, ensuring that the project benefits from our collective expertise and adaptability.**

**Scrum Roles**

1. **Product Owner:**
   * Represents the stakeholders and is responsible for defining the features of the application and prioritizing the backlog.
   * Ensures that the team is focused on the most valuable work by continuously refining and prioritizing the Product Backlog.
2. **Scrum Master (Yonatan Azaraf):**
   * Facilitates the Scrum process, ensures that the team adheres to Scrum principles, and removes any impediments that might slow down the team.
   * Conducts meetings and keeps the team on track to meet deadlines.
3. **Development Team:**
   * The development team consists of all members: Yonatan Azaraf, Rohan Chhika, Naftali Diner, Devesh Chiba, Shayur Govin, and Sayj Ramsamy.
   * The team is self-organizing and cross-functional, responsible for delivering potentially shippable increments at the end of each sprint.

**Scrum Events**

1. **Sprint Planning:**
   * At the beginning of each sprint, the team meets to plan the work to be done. We select user stories from the Product Backlog and break them down into tasks, estimating the effort required for each task.
   * Sprint goals are defined, and the team commits to completing these goals within the sprint duration.
2. **Daily Stand-Ups:**
   * Every day, the team holds a short meeting to discuss progress, identify any blockers, and plan the day's work.
   * Each team member answers three key questions: What did I do yesterday? What will I do today? Are there any impediments?
3. **Sprint Review:**
   * At the end of each sprint, the team presents the current release during the sprint to the Product Owner and stakeholders.
   * The team demonstrates the functionality developed, and feedback is collected to be incorporated into future sprints.
4. **Sprint Retrospective:**
   * Following the Sprint Review, the team reflects on the sprint process during the Sprint Retrospective.
   * The group evaluates what went well, what didn't go well, and where adjustments can be made. It is decided to take concrete actions to improve the process in the upcoming sprint.

**Sprint Structure**

* **Sprint Duration:** In our project, a sprint lasts one week, beginning on Monday and concluding on Sunday. This makes it possible to deploy functional components continuously and iterate quickly.
* **Sprint Goals and Backlog:** The Scrum team chooses user stories from the Product Backlog to establish the Sprint Backlog at the beginning of each sprint. Every sprint has a distinct objective, with an emphasis on delivering a certain set of additions or enhancements.
* **Backlog Refinement**
* **Continuous Backlog Refinement:** Throughout the project, the Product Owner and the development team work together to continuously improve and order the Product Backlog. By doing this, the team can make sure that they are constantly working on the most important items and that the backlog is current and relevant.

**Incremental Delivery**

* **Potentially Shippable Product Increment:** The team produces a potentially shippable product increment—a finished, functional section of the application—at the conclusion of each sprint. This guarantees a steady advancement of the project towards the ultimate output.
* **Adaptation and Flexibility**
* **Responding to Change:** Because Scrum is iterative, the team may adjust to shifting priorities or requirements. The backlog is modified based on input from sprint reviews to make sure the finished product fulfills stakeholder expectations.

**Communication Plan**

* 1. **Daily Stand-Up Meetings**
     + - Coordinate team activities and schedule the following day's work.
       - Locate and resolve any obstacles or problems.
  2. **Weekly Team Meetings**
     + - Examine developments, obstacles, and upcoming actions.
       - Review project milestones and update the progress.
  3. **Bi-Weekly Stakeholder Meetings**
     + - Get input from stakeholders and provide them updates.
       - Address any issues and make any necessary plan adjustments.
       - Ensure informed and involved stakeholders.

4. **Project Management Tools**

* + - * Use Jira for task management and tracking. Document and track progress, assign tasks, and monitor deadlines.

**Conclusion**

The goal of the UniEats dining hall app is to provide a functional dining hall application that satisfies the needs of our stakeholders. We pledge to deliver the solution on schedule by adhering to our project plan and utilizing agile concepts. The project's effective completion will be ensured by the communication strategy in conjunction with the remaining development plan components. Our objective is to offer a way to integrate the dining hall services into the Smart Campus program.