SW Engineering CSC648/848

“FoodsOnly”

Section 01 Team 06

Issac Moreno (Team Lead)

Noah James Yamsuan (Scrum Master)

Anshav Upadhyay Nepal (Front End Lead)

Terrell Enoru (Back End Lead)

Kayla Young (GitHub Master)

Karl Xavier Layco

Milestone 3

4/15/2024

Appendix I

Section: 01 Team: 06 "Foods Only" Date: 04/15/2024
Number of students present: 4

1. *UI and functionality feedback (P1 functions only)*

- **Instructor's comments on UI/functionality for your demo**

In the nav bar where is the blog. Choose whether to include it today. What is the biggest value that our application is providing? Discuss how we will finalize how machine learning will be used to customize the user experience.

- **Your Plan for the comments**

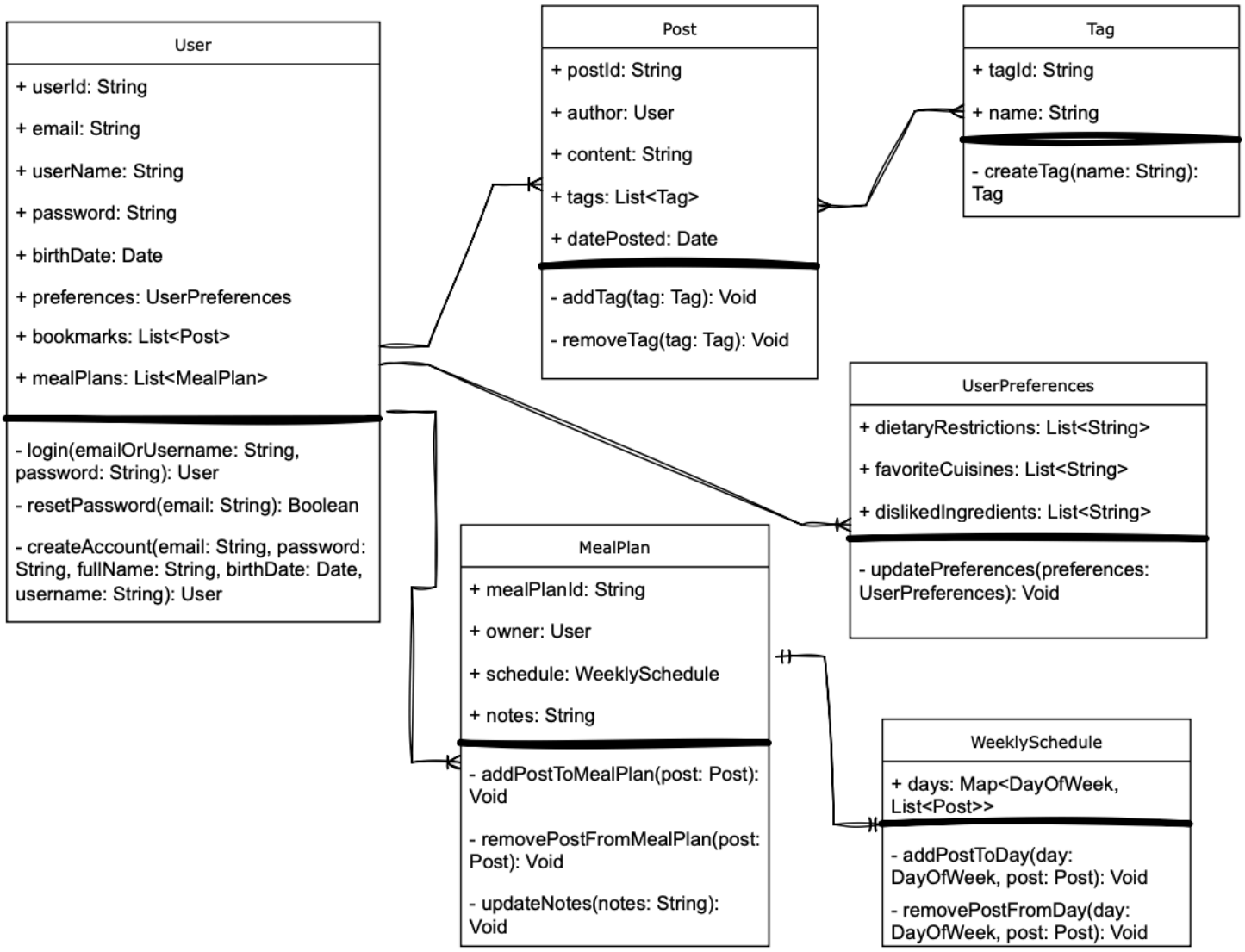
We have decided to not implement the blog and instead implement a restaurant service. Our biggest value is user customization using artificial intelligence to tailor recipes to our users. We will send the API the preferences and dietary restrictions of the user and then chatgpt would recommend recipes.

2. *List of P1 features committed for delivery*

- 1: Signing up
- 2: Logging in (sessions)
- 3: Searching (with AI)
- 4: Home page (Trending/personalized)
- 5: Profile customization
- 6: Posting

3. *Architecture*

[Next page]


User
+ userId: String
+ email: String
+ userName: String
+ password: String
+ birthDate: Date
+ preferences: UserPreferences
+ bookmarks: List<Post>
+ mealPlans: List<MealPlan>
- login(emailOrUsername: String, password: String): User
- resetPassword(email: String): Boolean
- createAccount(email: String, password: String, fullName: String, birthDate: Date, username: String): User
Post
+ postId: String
+ author: User
+ content: String
+ tags: List<Tag>
+ datePosted: Date
- addTag(tag: Tag): Void
- removeTag(tag: Tag): Void
Tag
+ tagId: String
+ name: String
- createTag(name: String): Tag
UserPreferences
+ dietaryRestrictions: List<String>
+ favoriteCuisines: List<String>
+ dislikedIngredients: List<String>
- updatePreferences(preferences: UserPreferences): Void
MealPlan
+ mealPlanId: String
+ owner: User
+ schedule: WeeklySchedule
+ notes: String
- addPostToMealPlan(post: Post): Void
- removePostFromMealPlan(post: Post): Void
- updateNotes(notes: String): Void
WeeklySchedule
+ days: Map<DayOfWeek, List<Post>>
- addPostToDay(day: DayOfWeek, post: Post): Void
- removePostFromDay(day: DayOfWeek, post: Post): Void

4. Project status

- The current status of our application development is on track. We currently have 5 major features with limited functionality. As major features, we have signing up, logging in (sessions), our home page, searching, and profile creation. These features are not 100% done, requiring more fine-tuning and debugging. However, they work as expected and serve as a prototype. Our team works together collaboratively while coding and ensuring that only stable code is deployed.
- **Risks**
    - **GitHub Instability:** Sometimes, accidents happen! A risk we may encounter is a GitHub branch deletion/merge that makes our application unstable. In this occurrence, we can rely on GitHub's ability to revert changes. But, it is not always guaranteed. Our plan to prevent these events relies on GitHub knowledge and communication while developing.
    - **Schedule Conflicts:** One of the highest risks we have encountered (and most likely will) is timing and scheduling conflicts. When our schedules do not align or a teammate cannot allocate enough time for the project, we may fall behind in development. This risk has happened in the past but we were able to keep on track. In the future, this risk may be larger due to our larger-scale development. Our solution is to plan ahead for such circumstances and be prepared to be a team player.
    - **Legal Risks:** If we were to deploy our application for public use, we may run into legal issues for things like data collection, API usage, and copyright issues. To prevent these, we could thoroughly read the terms and conditions for each component of our project used.
    - **Communication Risks:** Miscommunication between teammates is a major risk as well. If not handled correctly, we may cause a delay in development and production. Collaboratively working with 5 other teammates can be challenging, but our solution is to actively communicate. We mostly meet virtually, to our advantage, which promotes active participation. Additionally, we utilize communication platforms like Discord and Notion to send instant messages and access our current project details.

Appendix II

Section: 01 Team: 06 “Foods Only” Date: 04/15/2024

Instructor/TA to Check and comment below:

- Git/Github organization (e.g. organization of branches)
- Git/Gith, git hub usage: code review practices (to see if the review comments are proper and enough)
- Frameworks (back end front end) deployed correctly