Staff-relations management

Sprint 1: documentation

Claire Newman (2549861)

Nashita Chowdhury (2554397)

Dineo Chiloane (2563191)

Chloe Dube (2602515)

Mayuri Balakistan (2543986)

Kelly Wald (2552337)

1.1 Scrum methodology

Minutes for Meeting 1

Time: 4pm to 5pm

Date: 11 April 2024

Location: Google Meets Meet – sbt-zivh-ukc (google.com)

Project: Staff Relations Management

Atendees:

Claire Newman (2549861)

Nashita Chowdhury (2554397)

Dineo Chiloane (2563191)

Chloe Dube (2602515)

Mayuri Balakistan (2543986)

Kelly Wald(2552337)

Topics discussed:

Basic discussion of product

Created a shared word document to incrementally add ideas to the product.

Discussed points of project, such as how the login will look, and proposed colour theme. Undecided on colour palette (to ask client).

Created shared GitHub repo, as well as how to add use cases.

Discussed what will happen during Sprint 1: completing at least 4 user stories, working well through GitHub, assigning tasks.

Created a discord server to share resources.​

To set up meeting with client

We discussed the times in which meeting the client would work out best for all team members.

Best time for all members is 9am.

Sent mail to client to check if they too are available at the specified time.

When to do marking/sprint review(?)

Discussed which day would work best for the team to show the client the product and to mark it. Claire suggested Tuesday afternoon, and this was agreed upon.

Things to think about before meeting client:

Choosing architecture. Possibly Server-client.

Choosing best coding languages: JavaScript, HTML and CSS.

Choosing which cloud computing platform to use: Azure.

Questions to ask clients:

Are there any specific requirements (non-functional and functional)

Specific domain name?

What user stories must be implemented?

Should we make GitHub repo public?

Clients preferred design aspects- layouts, specific colour palette.

How to deal with use cases:

Get the use cases from the client.

Create product backlog.

Add to GitHub.

Update the refinement backlog.

break down user stories into smaller tasks.

Assign use cases to team members.

Overview of SCRUM:

Sprint planning- to occur after discussing use cases with the client, create product backlog.

Daily scrum- put time aside to complete and discuss progress.

How to deal with sprint review- possibly online if the clients schedule does not match up to the Team members’ schedules. We can share the screen with client to show WIP.

Sprint retrospective-may happen simultaneously with backlog refinement.

Backlog refinement- reassess product backlog and discuss how to deal with remaining user stories.

Items to be discussed next time:

Meet up with the client.

Discussing use cases with the client.

Discuss general look of the website with the client and any specific requirements.

Minutes for Meeting 2

Time: 7:30pm

Date: 12 April 2024

Location: Discord

Project: Staff Relations Management

Participants:

Claire Newman (2549861)

Nashita Chowdhury (2554397)

Dineo Chiloane (2563191)

Chloe Dube (2602515)

Mayuri Balakistan (2543986)

Kelly Wald (2552337)

Topics to be discussed:

Breaking down and delegating user stories.

What code coverage platform, CI/CD delivery platform, web app service and 3rd party database we will use.

How we will track user stories.

User stories given:

Login for manager, staff, and HR.

* As a staff member, I want to be able to login to access my profile.
* As an HR member, I want to be able to login to access my profile.
* As a manager, I want to be able to login to access my profile.

Password verification

* As a staff member, when I login with the incorrect password, I want it to notify me.

Design specifications:

* None were given.
* Dev team free to use own designs.
* Colour palette to be discussed at a later stage.

Login:

* A Home page button is required.
* Place for a username and password is required.
* 3rd party login database-Firebase
* Login must be able to differentiate between the different users.
* If a password or username is incorrect, user will be notified, and login will not work.

Code Coverage:

Code Cov:

* Can log in with your GitHub.
* Only free for a single user

Istanbul:

Does not integrate with GitHub.

Coveralls:

Only free for Open-Source projects.

* Integrates with GitHub.

Team members undecided on which code coverage platform to work with.

How to track user stories

We will use GitHub Projects.

Documentation:

Explain sections of code.

Allows code to be more understandable.

Include how to test certain sections of code.

CI/CD:

CircleCI:

* Free
* Integrates with GitHub

GitHub actions:

* Integrated with GitHub
* Free

Discussion of web app service?

Alternatives to Azure Static Web App:

* Azure App Services was an option.

The team members agreed to stick to Azure Static Web App.

Signing in to Azure student is a must.

Delegation of work:

HTML login and signup page: Mayuri

User verification using Firebase: Chloe, Dineo

Make Azure static web app: Nashita, Kelly

Password verification: Chloe, Dineo

Get CodeCov to work: Claire.

Set up GitHub actions/CircleCI: Claire

To be discussed at next meeting:

Whether GitHub Actions or CircleCI is a better choice for CI/CD.

Which code coverage platform will be used.

The progress of each team members’ assigned work.

Minutes for Meeting 3

Date: Saturday 13 April 2024

Time: 16:00

Location: Discord

Attendees:

Nashita, Dineo, Claire, Chloe

Absentees:

Kelly, Mayuri

Agenda:

1. What’s the plan for today.

2. What did we do yesterday.

3. impediments

Discussion:

• Continue with refining the user authentication – specifically with terms of account distinction (admin, manager, staff)

• Discussed what has been done with the logins (progress report)

• Discussed implementation of jest into the cl/cd pipeline.

• Discussed plans to finish UATs.

• Nashita Needs login code before launch to azure.

Decisions Made:

1. [Nashita will help with UATs] o [Lighten Claires Load

Next Meeting: [17:00, 14th April, Discord] Adjournment: [16:28]

Minutes of Meeting 4

Date: Wednesday 17 April2024

Time: 19:13

Attendees:

Claire Newman (2549861)

Nashita Chowdhury ([2554397](mailto:2554397@students.wits.ac.za))

Dineo Chiloane (2563191)

Chloe Dube (2602515)

Mayuri Balakistan (2543986)

Kelly Wald ([2552337](mailto:2552337@students.wits.ac.za))

1. Commencement of the meeting:

- The meeting officially started at 19:13.

2. Project Progress:

- No significant work has been done for the project so far.

- One member attempted to implement unit tests for the login functions using Firebase but encountered difficulties.

- Testing options discussed were online testing (requiring secrets) or offline testing.

- Issues with importing modules and configuring tests were encountered.

3. Documentation and Planning:

- One member plans to start working on documentation and UML diagrams the following day.

- The upcoming weekend is expected to be busy for some members, potentially affecting their availability.

- A request was made to schedule a mock session at 2:00 for Sprint One evaluation.

4. Work Assignments:

- One member volunteered to work on HTML pages and access-related tasks during the weekend.

- Unit tests are currently assigned to a member, but they expressed struggling with the task.

- Helped and suggested creating a separate branch for tests to avoid impacting the main repository.

5. Miscellaneous:

- Discussion about the limitations of using Jest for testing Firebase and the switch to Mocha.

- Challenges with imports and the need for module configuration were raised.

- Offline testing was preferred due to the sensitive nature of secret information.

- Agreement reached to create a separate branch for tests to avoid conflicts with the main codebase.

Meeting Adjourned: [20:00].

1.2requirement engineering:

functional requirements

1. HR administrators have full access to modify, remove, and add data regarding an employee or a manger. They also can change permissions for all user roles available.
2. HR admins can request feedback from staff members and specify recipients. Moreover, they will have access to view performance history for all members.
3. HR should be able to create meal options for staff members to book and they should be able to view all bookings made by the staff for meals and car washes.
4. HR have access to comprehensive reports made such as timesheets and performance feedback (per user or task). They can customize these reports based on specific criteria and make them exportable as pdfs or csv files.
5. Managers must have limited access to the system and are only allowed to modify employee data and change permissions for staff members within their assigned teams.
6. Managers can provide feedback to their team and have permission to view performance feedback given to them by their employees.
7. Managers will be able to view bookings made by their team members for company events like meals and car washes.
8. Managers will have access to the relevant report such as timesheets per employee and per task within their team.
9. Managers must be given the ability to generate reports based on specific criteria and said reports must be exportable in a csv or pdf file.
10. Employees must be able to record and track time they spent on different tasks in the timesheets given to them and they can enter and manage their timesheet data.
11. Employees can type out performance feedback sheets about their fellow employees, managers, and HR administrators.
12. Employees can make bookings for company events such as meals and car washes. They must also be able to view their booking.
13. Employees should have access to their own timesheet reports, showing their recorded hours for different tasks and projects.
14. All system users will receive notifications on relevant dates of their bookings.
15. Users must also get notifications for weather warnings if they have booked for that date.

Non-functional requirements:

1. Security: The application should adhere to industry-standard security protocols to ensure the confidentiality, integrity, and availability of sensitive employee data.
2. Scalability: The application should be designed to handle a growing number of users and data without compromising performance or user experience.
3. Usability: The user interface should be intuitive, user-friendly, and accessible to users of varying technical expertise, ensuring ease of navigation and interaction.
4. Performance: The application should have fast response times, minimizing loading and processing delays to provide a seamless user experience.
5. Reliability: The system should be highly reliable, minimizing downtime and ensuring that critical features are always available and accessible to users.
6. Compatibility: The application should be compatible with commonly used web browsers, operating systems, and devices to ensure broad accessibility across different platforms.
7. Data Backup and Recovery: The system should have robust mechanisms in place for regular data backups and efficient recovery procedures to protect against data loss and enable restoration in the event of system failures.
8. Compliance: The application should comply with relevant legal and regulatory requirements, such as data privacy regulations (e.g., GDPR) and employment laws, ensuring that employee data is handled appropriately.
9. Localization and Internationalization: The application should support multiple languages and cultural preferences, allowing users from different regions to use the system effectively and comfortably.
10. Documentation and Support: The application should be accompanied by comprehensive documentation, including user guides and system manuals. Additionally, there should be a support mechanism in place to address user inquiries, issues, and provide timely assistance.

1.3 Software Implementation, User Stories:

1. As a staff member, I want to be able to login to access my profile.
2. As an HR member, I want to be able to login to access my profile.
3. As a manager, I want to be able to login to access my profile.
4. As a staff member, when I login with the incorrect password, I want it to notify me.
5. As an employee, I want to be able to create an account if I am new to the company.
6. As an HR member, when I login with the incorrect password, I want it to notify me.
7. As a manager, when I login with the incorrect password, I want it to notify me.
8. As HR, if a user with an invalid account log in, I want them to be notified

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| UAT title | Description | Test steps | Expected results | Actual result | Status of test | input |
| Login on website as employee | Login to the website with a valid staff account | 1. Click the link 2. Type in a valid username and password 3. Press login button. | Login Successful popup | successful input. User taken directly to main page | pass | email: mazzy@gmail.com ,password: Password123? |
| Login on website as HR | Login to the website with a valid admin account | 1. Click the link 2. Type in a valid hr email and password 3. Press login button | Login Successful popup | Successful input, User taken directly to main page | pass | email: file@gmail.com password: File123? |
| Login as manager | Login to the website with a valid manager account | 1. Click the link 2. Type in a valid manager email and password 3. Press login button | Login Successful popup | Successful input, User taken directly to main page | pass | email: beyonce@gmail.com password: Beyonce1! |
| Login as employee with invalid password | Enter in a valid staff email and an invalid password | 1. Click the link 2. Type in a valid email and invalid password 3. Press login button | Login Failed popup | Failed input, user received message "Wrong email or password. Please try again." | fail | email: clion@gmail.com, password: 1234 |
| Login as manager with invalid password | Enter in a valid manager email and an invalid password | 1. Click the link 2. Type in a valid email and invalid password 3. Press login button | Login Failed popup | Failed input, user received message "Wrong email or password. Please try again." | fail | email: beyonce@gmail.com password: Beyonce |
| Login as HR with invalid password | Enter in a valid HR email and an invalid password | 1. Click the link 2. Type in a valid email and invalid password 3. Press login button | Login Failed popup | Failed input, user received message "Wrong email or password. Please try again." | fail | email: file@gmail.com, password: Red111 |
| Login to website with invalid account | Enter invalid email and password | 1. Click the link 2. Type in a valid email and invalid password 3. Press login button | Login Failed popup | Failed input, user received message "Wrong email or password. Please try again." | fail | email: 333@gmail.com password: wedIO0 |
| Create staff account | Create a staff account | 1. click the link 2. Select create an account. 3. Type in name, surname, email and password. 4. click sign in | "Your account was successfully created. Go back to the sign in page and sign in." text pops up | User was shown the message: "Your account was successfully created. Go back to the sign in page and sign in" | pass | name: Name, Surname: Sname email: NewUser@gmail.com password: NewPassword , access key: uR789789 |