**APPOINTMENT SCHEDULER APPLICATION PROJECT**

**Developer Meeting Report**

**Meeting Date:** 27/03/2024

**Participants**

|  |  |
| --- | --- |
| **Customer Team** | **Developer Team** |
| Hatice Sebla Karabunar | Abdel Aziz Mounther Qudeih |
| Ali Berke Devecioğlu | Ahmed Alidlbi |
| Tahsin Emre Telli | Mohamad Malek Alkhodary |

Haidar Kahla

**Objective**

To discuss requirements and expectations for the development of the Appointment Scheduler App with the development team.

**Key Discussion Points**

**Project Overview:**

The customer team convened to initiate discussions with the development team regarding the creation of an Appointment Scheduler App. We conveyed our vision for the app and sought alignment with the development team's capabilities.

**Feature Prioritization:**

Outlined the primary features essential for the Appointment Scheduler App:

1. User registration and login

2. Appointment booking and management

3. Notification reminders

4. Integration with calendar systems

**Technical Requirements:**

Highlighted key technical considerations including:

1. Platform compatibility (iOS, Android)

2. Database management

3. Security measures for user data protection

4. Multiple language support (English and Turkish)

5. Notifying if two meetings overlap

6. Sending an email to participants when an appointment is scheduled

**Project Timeline and Milestones:**

Agreed upon a preliminary timeline featuring milestones such as:

1. Requirements gathering phase

2. Design and prototyping phase

3. Development phase

4. Testing and debugging phase

5. Deployment and launch

**Assigned Tasks:**

1. Feature Definition:

- Provide detailed specifications for each feature identified.

2. UI/UX Design Preferences:

- Share preferences and expectations regarding app design and user experience.

3. Integration Requirements:

- Specify any third-party integrations required, such as calendar systems.

4. Feedback Mechanism:

- Establish a framework for providing feedback and reviewing progress throughout the project.

**Next Steps**

1. Finalize feature specifications and share with the development team by the next meeting.

2. Schedule regular progress updates and feedback sessions to ensure alignment with project objectives.

**Next Meeting Agenda:** 03/04/2024

UI/UX DESIGN /

The client has reviewed a portion of the UI/UX design, and we are currently evaluating the remaining sections.

Figma Link : <https://www.figma.com/file/Wr8rErBIEdVevL1x9YqOxR/Hora?type=design&node-id=10%3A2&mode=design&t=D4AahTYr8ADz43Ij-1>

TECHNOLOGY STACK:

MERN stack

React :

We use ReactJS for its efficient, flexible, and reusable components, which streamline the development of interactive user interfaces.

NodeJS:   
We use Node.js because it enables scalable, server-side applications and efficient data-intensive real-time functionality across distributed devices.

ExpressJs:

Express.js is used for its minimalist structure that simplifies the development of robust web applications and APIs with optimal performance and routing capabilities.

MongoDB :

MongoDB is used for its flexible schema design, which allows for easy storage and retrieval of large volumes of structured and unstructured data in a highly scalable fashion.

DIAGRAMS :

We create diagrams for our applications to visually represent system architecture and workflows, enhancing understanding and facilitating better communication among team members and stakeholders. These diagrams also aid in identifying potential issues and streamlining the development process.

Use Case diagram:

Use case diagrams visually represent user interactions with a system, clarifying requirements and functionalities.

Une image contenant diagramme, croquis, dessin, ligne

Description générée automatiquement

Use case diagram

Activity diagram:

Activity diagrams are used to model the workflow and sequence of operations in a system, showing the dynamic aspects of information flow and decision paths.

Une image contenant croquis, diagramme, texte, dessin

Description générée automatiquement

Activity diagram

Class diagram :

Class diagrams are used to visually represent the structure of a system by illustrating its classes, attributes, operations, and the relationships between objects.