The requirements were elicited through two main methods. Firstly, we faced a personal need for a mobile scheduling application and thought about what features would be most beneficial for us. Secondly, we looked at an already existing scheduling application, Doodle, and figured out what it’s functionalities were and what features our product would also need.

Problem Description:

Our product is a mobile application that helps users schedule meetings. It is designed for teams and other small groups in both business and casual settings. It streamlines the scheduling process by handling the recognition of overlapping available times, and eases the decision making process. The original alternative to our product was emailing everyone involved and trying to figure out availability through back and forth communication. This process is inefficient because each person generally only know their schedule. Our app will present available meeting times without needing discussion and debating.

The objective of our project is to allow users to make a group in the app, and then send notifications to group members stating their desire to schedule a meeting within the next week. Then, users indicate their availability at various times on each day within the period. Users can designate a time as unavailable, available, and partial availability (this can include an undesired time or a virtual presence in an in-person meeting). After the app determines the overlapping times, ranks them by percentage of participants that are fully available, then by partial availability. Users then vote for a time. We will keep the scope to scheduling a meeting within a one week period because setting availability for more than 7 days would be tedious, and schedules are often much more variable outside of that time.

Our main competitor is Doodle. Doodle’s approach is to allow hosts of events to send times to group members and have them take a poll on which they prefer. Our approach will be to have the app find availabilities for the user, which will provide options that a host could possibly miss. In addition, we will have our app will adapt to emergency schedule changes automatically. When a user changes availability on an already voted on event, a message will be sent to the group leader asking if the group should vote again. Then, the app repopulates the list with the overlapping times, and the group decides another time. Another feature that will differentiate our product will be an agenda feature that implements image recognition technology to translate handwritten notes into digital notes for the meeting.

User Functional Requirements

1. The group leader invites group members to attend a meeting within the next week via a notification sent by the app.
2. The users are then asked to input their availability on each day within that week. The users can designate stretch of time within that day as unavailable, available and as partially available. When a user is marked as partially available, they are asked to specify whether the time is undesirable or they cannot physically attend but can virtually attend.
3. The product calculates overlapping times from the given data and ranks the overlapping times by percentage of availability by the group.
4. Users then vote on the times in order to determine the best time for everyone.
5. If a group member has a change in availability, a message will be sent to the group leader asking if the group should vote again. Then, the app repopulates the list with the overlapping times, and the group decides another time.
6. Hosts can create an agenda for the meeting by handwriting it and the app will convert the handwriting to digital text, or they can type it in directly.