

Whispper Project

- Initial Report -



Fernando Vilma
Nicolas Cabuli

Khalid Alobaid
Oluremi Obolo

Mariya Abdiyeva
Santiago Arrubla

7 Febraury 2017

1 Project Description

1.1 Requirements (Aims)

Our aim is to create a distributed chat system with the following specific requirements by priority.

1.2 Strategy and Timetable

Our strategy is to develop three client-based platforms (i.e. Android, iOS and Web) using Firebase as a server. The reason for including a third platform is to act as a backup should there be problem with any of the platforms during testing which we are unable to debug at the end. The tentative timetable (Figure 1) presents the main activities required to complete our final product. It is important to state that the timetable and our strategy is constructed based on the Scrum [1] methodology, which is an agile software development guide that allows us to iterate and make incremental and evolutionary contributions. We will also use different Project Management roles like a Scrum Master (who removes team obstacles and pushes resolutions) and the product owner (who acts as the client voice) to ensure a high-quality product. The main activities are explained as follows:

1. **Plan:** Here we defined how we are going to work, our goals and organization as a team. We also discussed our individual strengths and weaknesses and the tools available to achieve our aim and objectives.
2. **Firebase:** For simplicity, costs (free) and security we are using this Backend as a Service platform to have a robust system which takes care of the authentication, rules for accessing the database (read/write) and the storage of documents so that we could focus on the core functionality, i.e. the actual messaging.

Table 1: Requirements by priority level

Priority 1	Priority 2	Priority 3
-Secure connection -Encrypted messages -IOS, Android and Web platforms (at least two platforms) -Contact list -Search for contacts -Sign in by email or Gmail -One to one conversation (text message)	-Send images -Group chat -Chat bubbles instead of user-name -Notifications of received messages	-Sign in by Facebook -Availability to send other type of media (e.g. video, voice notes, etc.) -Online status of other users -Acknowledgment of received, delivered and read messages, including date time -Ability to use the chat offline

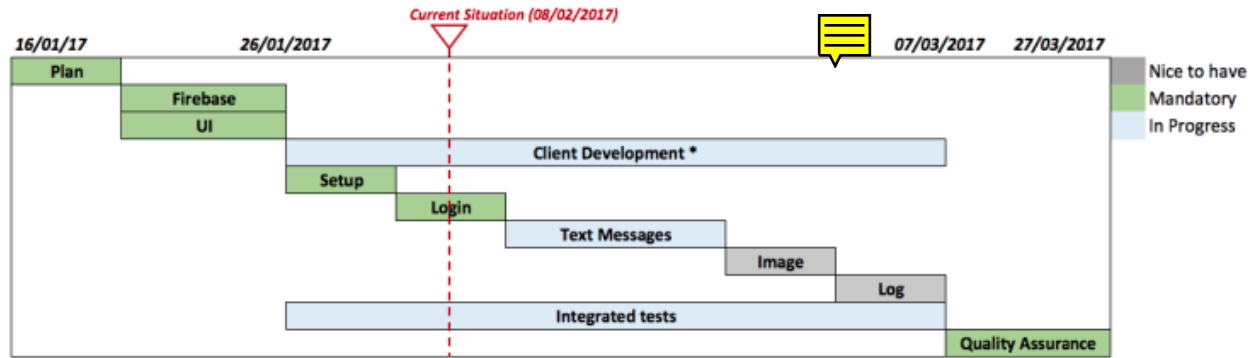


Figure 1: Timetable

3. **UI:** Refers to designing the User Interfaces for all the clients.
4. **Client Development:** This is the development of the specific application for each type of client (Android, iOS and Web):
 - (a) **Setup:** Here we add the required files and initialize the repository on GitHub. Then we create the project on Firebase so that we could share the same API Keys.
 - (b) **Login:** Build the login page, authentication using email and password, Gmail and Facebook if possible.
 - (c) **Conversations:** Text messaging between clients using Firebase as a central database containing the messages between recipients and taking care of the encryption. Add chat bubbles, send and receive messages, show them in an organized manner with timestamps.
 - (d) **Image Message:** Share images inside messages (nice to have).
5. **Integration Testing:** Test functionality across platforms.
6. **Quality Assurance:** Check that all clients communicate with the backend as intended in order to ensure that we cover all the required functionalities and fix any problem that we may have.

Please note that our activities in the client development are organized by hierarchy over time. Thus, developing the text message functionality (which is mandatory) is more important to us than the image message functionality (which is optional but highly desirable). Consequently, we decomposed our activities into tasks and allocated them to each team member as follows:

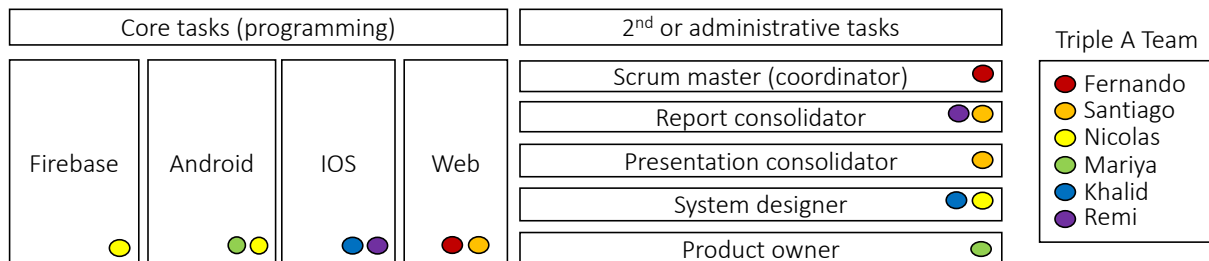


Figure 2: Tasks allocation

1.3 Current State

Currently, we have developed the authentication functionality inside the three clients. Firebase authorizes the logged users and allows them see the data only if they are logged in.

1.4 Next Steps

Following our timetable, we are going to focus on making the proper database structure (chats, messages, users, receiver, sender, type of message, etc), rules for accessing the data inside the database (only user x and user y can see their own conversation) and adding new data to the database.

2 Project Organization

In this section, we give more details about our organization, including how we will make decision and resolve conflicts.

2.1 Roles

Every team member has two types of tasks: core tasks (e.g. programming goals) and administrative tasks (e.g. documentation and updated). Additionally, the coordinator is the official team communicator. However, it is clarified that this role does not imply a position of leadership or decision making (the decisions are jointly made by following the decision making and conflict resolution methodology in Section 2.4). The specific roles were already stated in Figure 2.

2.2 Tools for Communication and Collaboration

- Slack (for communication purposes and specific tasks).
- GitHub (for code sharing and version control).
- Compulsory weekly meeting every Monday (12:00 PM - 2:00 PM).
- Compulsory Google Hangout every Sunday (5:00 PM- 6:00 PM).
- Extra Google hangout meeting (when needed).
- Meeting on Wednesdays 2:00 PM- 5:00 PM (if needed).
- A timetable and advance curve control (i.e. where we are vs where we ought to be).
- Our decision-making and conflict resolution methodology (Section 2.4).
- Learning meeting (each 15th day we discuss what we can improve as a team)

2.3 Peer Assessment Criteria

Our assessment methodology is based on the following set of criteria.

Table 2: Assessment Criteria

Criteria	Description
Amount of work	Amount of work that each member contributes (e.g. difficulty or complexity in the sub-set of requirements achieved by the member).
Team work skills	i) communication skills, ii) cooperation (e.g. how a member supports others) and willingness to resolve conflicts or reach agreements.
Proactiveness	Self-motivation to explore and propose new alternatives or options to overcome any difficulty. These include initiative, willingness to learn, good attitude to explore new things, enthusiastic, creativity and innovation.

Each member evaluates others on a scale of 1 and 5 (where 1 is the lowest and 5 is the highest score). For each criterion, each member would write a little justification, which could be replied by the other member. After-wards, the evaluator could decide to modify his assessment or otherwise. The final score of each member will be the average of all the valuations, weighted by the weight of each criterion (i.e. 1/3). Then, the 100 points would be proportionally distributed to meet with the assessment criteria of the module.

2.4 Decision Making, Agreements and Conflict Resolution Methodology

Making agile agreements and rapidly resolving conflicts are key factors towards maximizing our team output. Our methodology is inspired by the agents and multiagents theory [2], specifically in a practical reasoning agent BDI (Belief Desires - Intentions). In this manner, every team member possesses a valid point of view. Any difference between opinions goes through a deliberation process (where personal conflict is a conciliation process, which is a special type of deliberation), that would transform a set of options or desires to a set of intentions (this implies that each time there is a deliberation process, the first step is to state a set of options). Then we will plan and assign resources. However, the deliberation process required a balance due to time restrictions (based on our timetable and curve advance). This implies that we must state a mechanism to accelerate the process wherever it is appropriate (deliberation process output is the ideal, but practically we would not always have time for a full consensus). Also notice that we can modify our plan due a belief update. Each time we are unable to reach consensus due to time restrictions, we have specified an agreement protocol that precipitates a final decision where the group will move forward. For this, we use the *Borda Count vote procedure to try to maximize the preferences of the entire group.

Finally, for a personal conflict (such as a communication problem), there will be an arbitration protocol to resolve it. Basically, the team identifies those members that are not involved in the conflict and implore them to mediate. If after a conciliation process there is no agreement, referees will make the decision (only if there are 2 or more referees). If a referee abstains, then we would activate the vote procedure. The following flowchart summarizes our methodology.

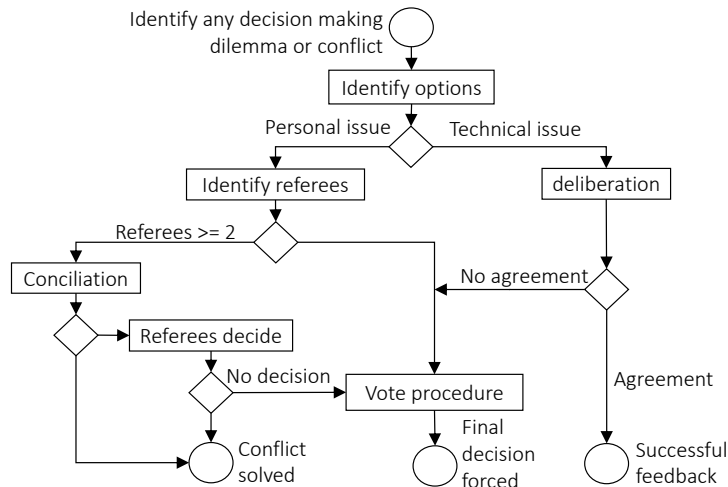


Figure 3: Decision Making and Conflicts Resolution Methodology

References

- [1] Cohn M. Succeeding With Agile-Software Development Using Scrum. *Pearson Education: Boston*, 2009.
- [2] M. Wooldridge. An Introduction to Multiagent Systems. *Department of Computer Science, University of Liverpool, UK, John Wiley Sons*, 0-471-4969 I-X, 2002.

*Borda Count: Each option receives $(m - 1)$ points for each voter who vote as a first choice, $(m - 2)$ for each voter who vote as a second choice, and so on, where m is the number of options. The winner option is that with more points. If there is a tie the decision will be randomly.