QuestCalendar

Ignacio Navarro Blázquez
Software engineer student
Hochschule Konstanz - HTWG
Konstanz, Germany
inavarroblazquez@gmail.com

Jade Neav
Computer science student
Hochschule Konstanz - HTWG
Konstanz, Germany
jade.lea.neav@gmail.com

***Abstract*—QuestCalendar is a mobile application developed on Java using Android Studio and implementing Firebase features. QuestCalendar allows the users to organize and improve themselves by gamification.**

Keywords—Java, Android, Android Studio, Calendar, Game, Health, Tasks

I. Introduction

QuestCalendar is an app developed for android devices, that aims at helping people improving themselves at the same time they organize their daily life. Created to focus on mental and physical health, social life, environment, and organization, based on the gamification of the things method, the user will be able to level up and gain experience by completing daily quests, and unlocking new characters or features to keep to motivation, and encourage de user to improve step by step.

II. Goals

Goals can be separated in different categories: organization, improvement, engagement, and satisfaction. But the main goal of the project is to create an app that people use in their daily life, adding tasks or events while users feel that they are playing a game.

A. Organization

QuestCalendar is a Calendar-based app, so users can organize their events and To-Dos on our app, in an easy way, and check them in a monthly or daily view, to give the users a overall view of what they have to do and reminding them their events by notifications.

B. Improvement

The application provides advice about different subject of improvement via notifications that you will receive during the day. The tips are easy to understand and follow. The different fields make each user to focus several fields to please every user. Each piece of advice is given with his source for more transparency.

C. Engagement and motivation

To help the users, it is important to enhance and feed their motivations. One quest is given to the user every day so they can feel having a small success in their life and congratulate themselves. Doing the daily quest make the user gain experience and level up. The user can also customize his character so give him another purpose to complete the quest other than a personal purpose.

Gamification is one technique that is proved to work and make happier the users creating them the necessity to beat the challenge or just fun, to make the day more easy-going.

D. Satisfaction

QuestCalendar aims at being stress-relief for the users, using light colors and a simple screen design to avoid creating them stress or anxiety by seeing a lot of information in one screen.

III. Requirements Analysis

For QuestCalendar, it has been created from personas and use cases to user stories and requirements that the application should satisfy.

A. Personas

- Miguel: He is a Full Stack programmer working for a big company. He spends all the time thinking about his job and lose the sense of time, also he is a gamer and in his free time he like playing videogames. He wants to focus on his job but also to focus on his personal life, improving his personal relations and health because he spends most of his time sitting in his chair. Miguel would like to be able to do some exercise in his break time of working to avoid back problems. He said, “I want to have more organization in my life, don’t forget things I have to do and fight laziness like if I were playing a game”. Some of his Key characteristics are:
    - Lose sense of time.
    - He thinks about his job all the time.
    - Always in his chair.
    - Likes playing games.

  And some of his goals:
    - Do exercise.
    - Improve himself and his relations.
    - Be focus on what he’s doing.
    - Don’t forget important things.

One use case scenario for him would be:

Miguel needs to take some pills every day at 9 am, but he is working since 7 am and he always forget it. He wants to get a reminder when he must take it.

Fig. 1. Some characteristics of Miguel

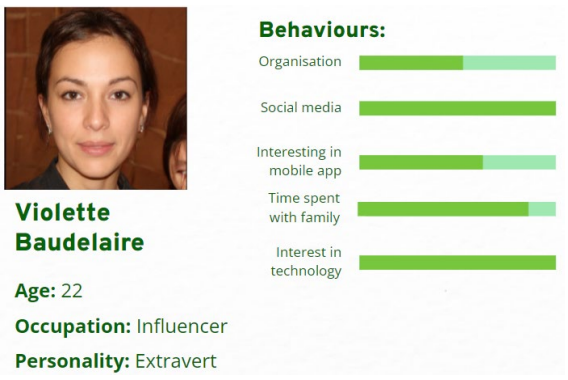

Fig. 2. Some characteristics of Violette

- Works as a youtuber and an influencer
- is passionate about science
- loves to take care of her family
- has not found a way of organization that fits her

And some of her goals:

- finding a system of organization that she can not lose.
- being able to share her way of organizing with her followers.
- having her personal and professional planning in the same space, but not mixed.

One use case scenario for her would be:

Violette has to post a video on Monday and to take care of her siblings the whole week-end. She needs to plan activities for her siblings but also the steps of the making of the video. She has a notebook that she uses as a planner but she is afraid of forgetting it or that her siblings will destroy it.

- last year of high school
- horse rider
- always anxious

And some of her goals:

- stop stressing to be able to study
- entering a good university
- having a good study planner

One use case scenario for her would be:

She wakes up and doesn't have energy to do anything, but she has an exam next week and she must prepare for it. She won't do it on the weekend because she has a horse-riding competition. The app makes her start working again by the daily tasks, which help her feel well by achieving small tasks.

B. *Storyboards*

There are also some storyboards created to simulate situations that could happen with and without a person using QuestCalendar.

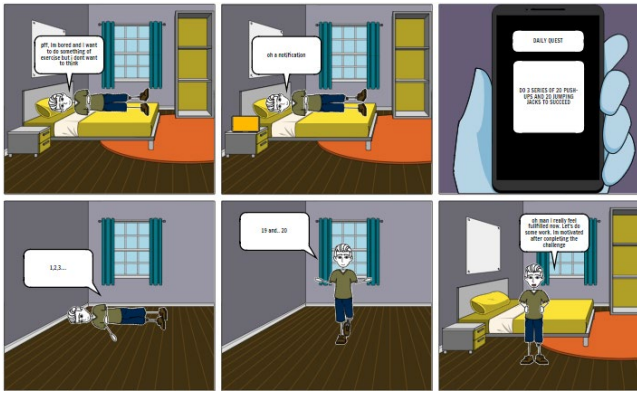

Fig. 3. Storyboard of doing exercise thanks to QC

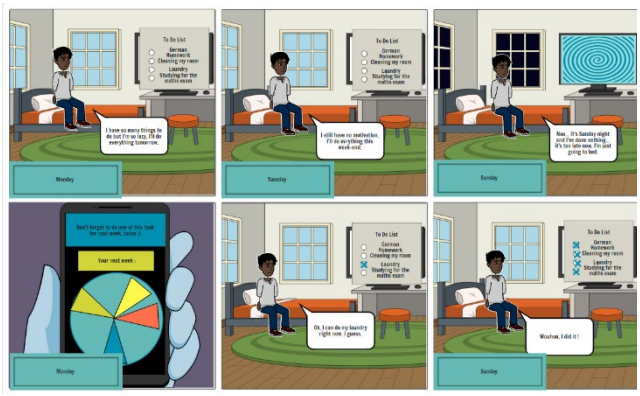

Fig. 4. Storyboard of organization thanks to QC

Fig. 5. Storyboard of healthy eating thanks to QC

Fig. 6. Storyboard of focusing thanks to QC

C. *User stories*

Number	User Stories
1	As a user, I want to create daily, weekly, and monthly events so that I can organize my schedule and don't forget anything.
2	As a user I want a beautiful interface because I want to enjoy the views when I'm using the app.
3	As a user I want to get all kinds of rewards so I can keep my motivation and don't forget to do my tasks.
4	As a user, I want to get notifications from the app because I don't want to forget events.
5	As a user, I want to be able to see my achievements because I want to keep track of my progress.
6	As a user, I want to receive one random quest every day because I want to keep my motivation.
7	As a user, I want to be able to check a task to be sure I finished it and receive experience.
8	As a user, I want to personalize my character so I can enjoy the application more.
9	As a user, I want to gain experience and level up by completing tasks and quests to unlock more accessories for my character.
10	As a user, I want to be able to check the source of a tip because I need to trust the application.
11	As a user, I want to receive tips with healthy information so I can be more informed and enjoy the app more.
12	As a user, I want to edit my information because I can change my mind about what I want.
13	As a user, I want to create an account so I can use the app.
14	As a user, I want to delete my account to delete all the records in the database.
15	As a user, I want to see my tasks represented in the calendar, so I know what I have to do every day/week/year.

D. *Functional Requirements*

Functionals requirements are divided in sections depending to which feature they belong.

Login/Register:

-After downloading the app, the system should ask for login or registration to identify the user.
-The username must be more than 4 letters and less than 8 letters for the registration.
-The username must not be empty.
-The username must be unique.
-The password must be more than 6 letters.
-The password must not be empty.
-The password can only contain numbers and letters.

Calendar:

-The calendar page should show daily, weekly, and yearly views.
-The application should let the user create new events.
-The event must not be empty.
-The event must have a name, date, and periodic state.
-The event's name must be more than 3 letters and less than 25.
-The periodic state of the events can be none, daily, weekly, yearly.
-The application should let the user delete events.
-The application should let the user create new tasks.
-The application should let the user mark as finished a task.
-The advice should include a link with its reliable source.
-The application should send one quest to the user every day.
-The application should let mark as completed a quest.
-The quest that the user receives must be random.
-The application should let the user choose to receive notifications or not.

Profile:

-The profile page should show information about the user and option to modify it.
-The profile page should let the user delete the account.
-The character customization should show every asset the user has, for hat, chest, and legs.
-The character customization should save the appearance of the character of the user.
-The profile page should show the level of the user.
-The profile page should show the user's achievements.
-The user should unlock assets for the character when leveling up.

E. *Non-Functional Requirements*

- The application should look beautiful and stress-relieving.
- The application should be easy to use.
- The application should be fast.
- The application should have fast access to the database.
- The application should be able to send notifications.
- The application should be available in every moment.
- The application should run in every android.
- The application should support updates.
- The application should be in English.
- The application should give good advice.
- The application should be positive.

IV. Conceptual Model

For developing the app, it was decided to create the mockups and the UML diagram of what each class should save. The mockups were done on Figma, so it is also a flux or sequence diagram interactive that can be tested on references section.[1]

A. *Mockups*

This are the group of mock ups created for QuestCalendar, which are fitting quite well the final version.

The login and welcome page are quite similar to the final version, but it was decided to change a bit the logo and add some more text.

Fig. 7. Welcome page of QuestCalendar

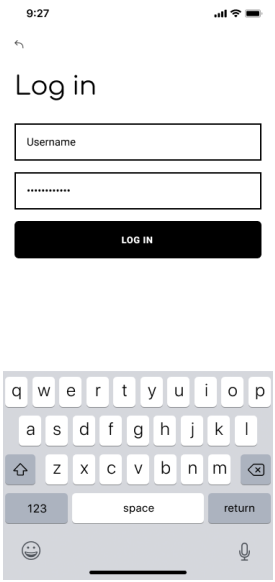

Fig. 8. Login page of QuestCalendar

The register section on the mockup was divided in two but we finally decided to put all in one page.

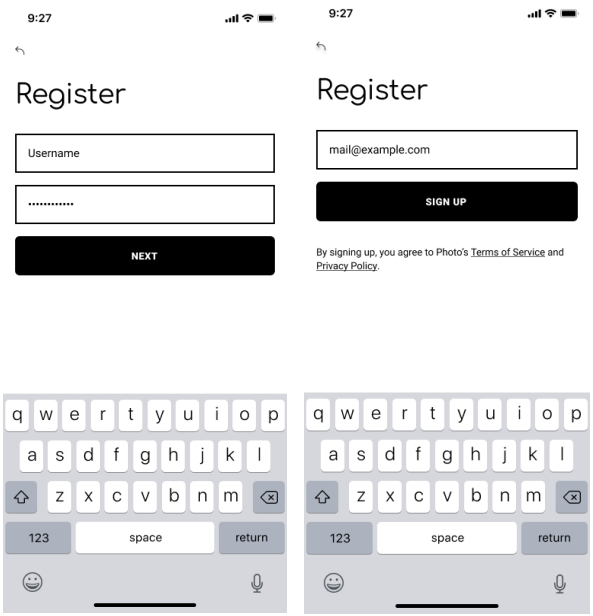

Fig. 9. Register page 1 of QuestCalendar

Fig. 10. Register page 2 of QuestCalendar

This is the profile and edit profile page, very similar to the final version, which cointains the characters and all the buttons.

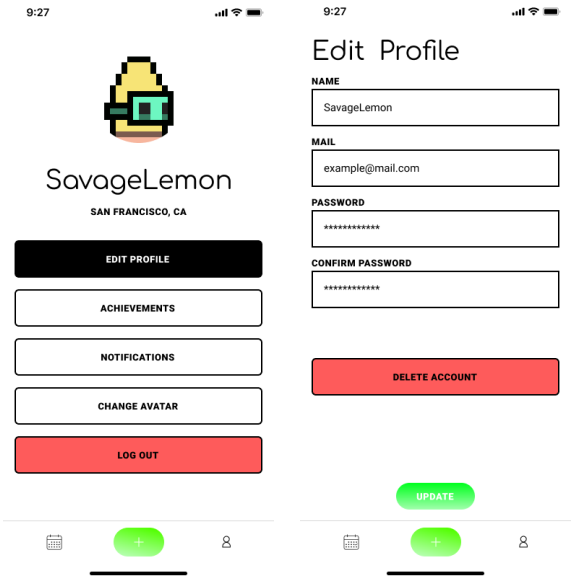

Fig. 11. Profile page of QuestCalendar

Fig. 12. Edit profile page 2 of QuestCalendar

For the avatar, our idea was to implement the personalization of the characters but finally it was not possible, anyway is similar to a character selection page, and the notification system is similar to the mockup.

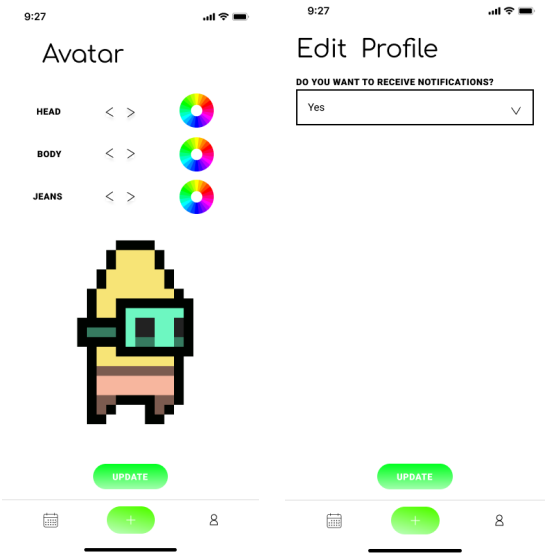

Fig. 13. Avatar selection page of QuestCalendar

Fig. 14. Notifications page 2 of QuestCalendar

This are the calendar view. Our intention was to have daily, weekly and monthly views. And the button of the daily Quest.

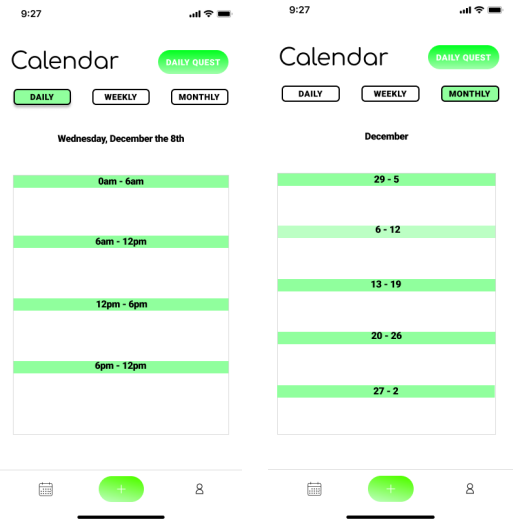

Fig. 15. Daily view page of QuestCalendar

Fig. 16. Monthly view page of QuestCalendar

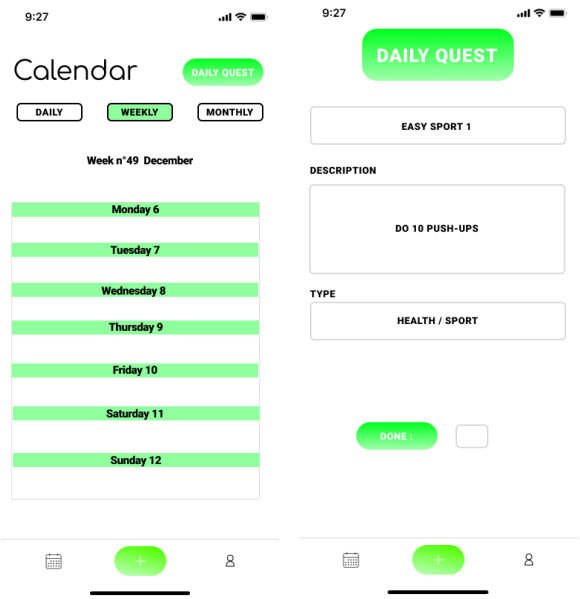

Fig. 17. Weekly view page of QuestCalendar

Fig. 18. Daily Quest page of QuestCalendar

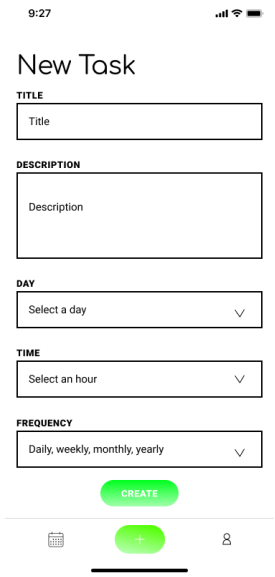

Fig. 19. Add Task page of QuestCalendar

B. *Diagrams*

As explained before, the only diagram is the UML one, because the sequence or flux diagram is integrated with the mockups.

The UML diagram represents the entities that we have created and that we are saving on Firebase Database.

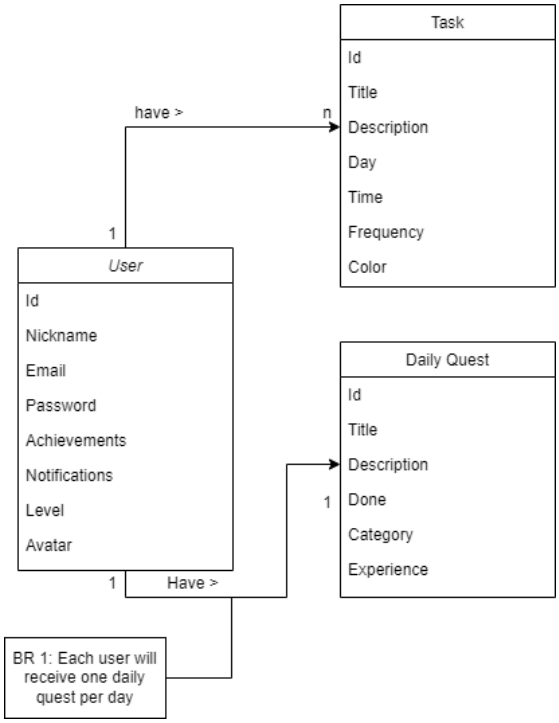

Fig. 20. UML diagram

V. Design Decisons

The main decision took was to have a nice-view app, simple and beautiful that creates a good ambient for the user, because the UI and UX are one of the most important things on applications. If you have a great app but with not a good experience for the user and a simple interface, you will not create the engagement necessary to keep users using your app.

A. *Bottom Navigation*

The first decision we take was to have a bottom navigation, with the three main functionalities of the application, because more would have made the users overwhelmed of information. Three simple sections: Calendar, add task and profile. This decision was very important because it's making QuestCalendar work with one main activity and fragments so information is shared differently than it would be from activity to activity.

B. *Firebase*

This is also an important decision because this is how the information of the users will be stored and would be hard to change on the future. It was decided to work with Firebase because it provides a simple way of storage like a tree, making it simple to access the information and updating it. But also, because it provides a good way of controlling the authentication system, in this case via mail

and password, giving also a userID autogenerated. Working with the auth it's making easier to organize the tree of the database.

C. *Calendar view fragments*

It was chosen to use two fragments called by the calendar fragment to display the daily view and the monthly view of the tasks. These views are displayed by pushing a button. This way, it will be easier to add the other views (daily and yearly) because it is just needed to add the respective button on the calendar fragment. Moreover, the fragments can communicate easily with each other using the calendar fragment. Besides, the user can access the Daily Quest from both views.

For the monthly view, the application displays a CalendarView. For the daily view, the application displays a list of the tasks of the day using a ListView. As the final goal of the application is to display the tasks in a colorful and original way, we chose to keep the display of the tasks simple to have an overview of the tasks in the first version of the application.

D. *Customizing avatars*

The first idea was to implement a character customization system, but we had more important functions to implement before this one, so it was decided to create a few characters that users will be unlocking while using the app, so they can choose between a selection instead of creating their own, that will have cost a lot of effort and time.

E. *Level System*

The level system was not a very prioritized user story or requirement, so when the moment came, we decided to make the following algorithm. The base experience needed to level up would be 50, but every time you reach a new level it's getting harder by the level squared. So, if a user is at level 1, it only needs 50 + 1^2 that is 51. And if the user is at level 20, it will need 50 + 20^2 equal to 450. With this system, it is not very difficult to level up until level 20, then you will need a lot of time to continue, but the idea was to make things unlockable until level 20, so after that you can continue leveling up but with no more rewards.

F. *Daily Quests*

Daily quest was supposed to be one each day, but because of testing and the question of how-to storage the time, it is only simulating one each day. Basically, when you mark the Daily quest as done, the button will be disabled for 24 hours. But for testing reasons, it was made so if you change the activity and return, you will be able to do another daily quest.

G. *Achievements*

It was planned to add an achievements section but due to time, we decided to implement other functionalities before, and on the near future add to the database a section for achievements and the activity for seeing them

H. *Tasks*

The tasks are saved in the database under the child *tasks* from the user ID. Tasks are displayed in the Daily View Fragment. To easily find the tasks related to the day selected by the user, a Task Manager is sorting the tasks according to their periodicity. Then computing the sorted list by hour of the task of the selected day.

A Spinner was used to get the date and the hour for the tasks. This way, the app ensures that the user is entering a valid date and a valid hour, avoiding problems on the database or by passing it to other functions.

It was also chosen to use Toggle Button so the user can't choose more than one periodicity for the task, making it punctual if the user doesn't choose any option, daily or monthly.

VI. Results

A video with the MVP of the application it has been shown on the presentation and can be found on references.[2]

A. *Evaluation*

For being sure that QuestCalendar is running smoothly with no crashes or bugs, the application has very good exception controlling system.

For example, each view that has a form on it, before sending the form, it must pass a hard validation form, not only with the minimum requirements but checking that no other data is being send like scripts or similar. And it's being recognized and letting the user know where and what is the failure.

The same with fragments and views, if something is sending exception, for example null pointer or wrong type, it will log it so it will be easy to identify it and manage it.

B. *Degree of completion*

The final degree of completion of the app in this moment is around 80% It is satisfying all but one of the non-functional requirements, all the login requirements, four out of seven of profile requirements and 11 out of 14 of the calendar requirements. Moreover, we added some extra functionalities, so the missing ones are only the ones related to notifications, and achievements and tasks. If it is counted with all new functionalities, the percentage would rise close to 90% but the ten percent missing are minimal functionalities that can be added on a near update.

C. *Development*

Here are some data of our work:

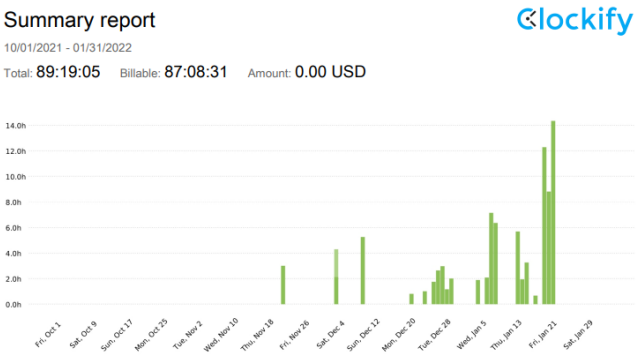

Fig. 21. Hours worked from oct. 1st to 31st jan.

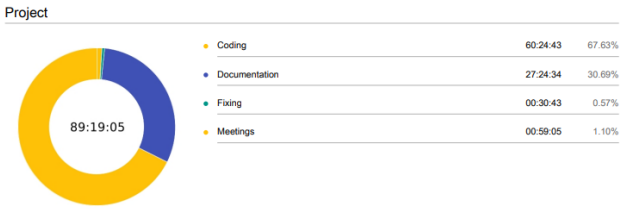

Fig. 22. Hours used on each task

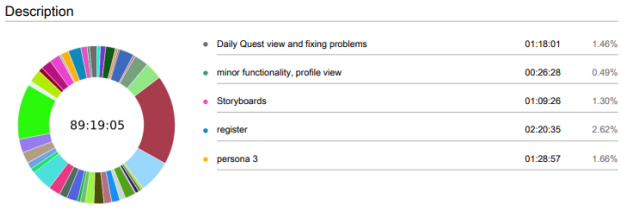

Fig. 23. Tasks separated

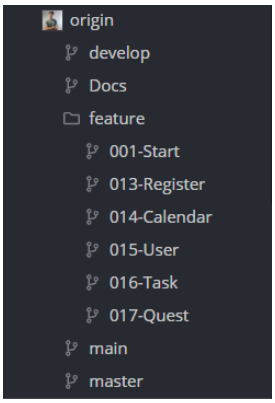

Fig. 24. Branches

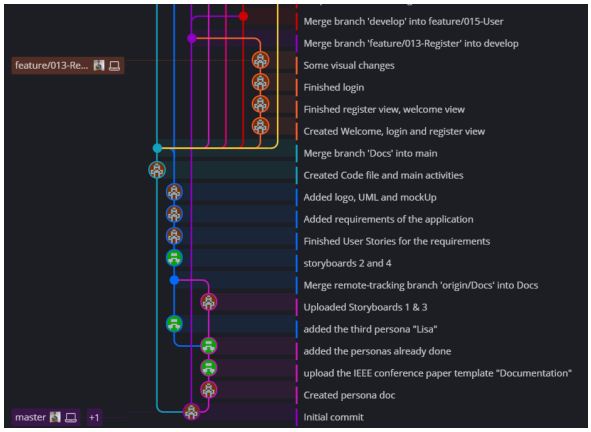

Fig. 25. Gitflow branch strategy being applied

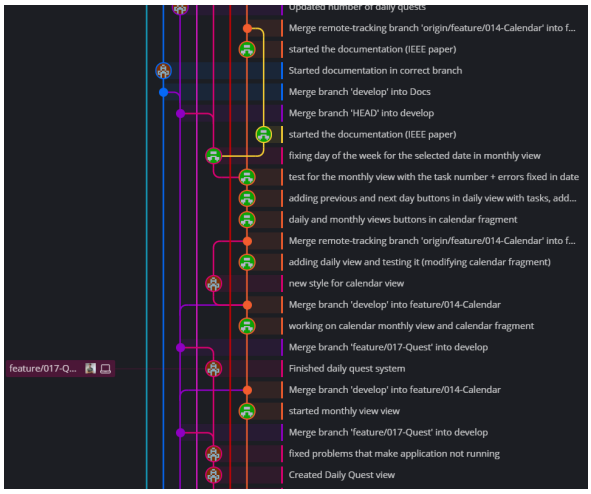

Fig. 26. Gitflow branch strategy being applied

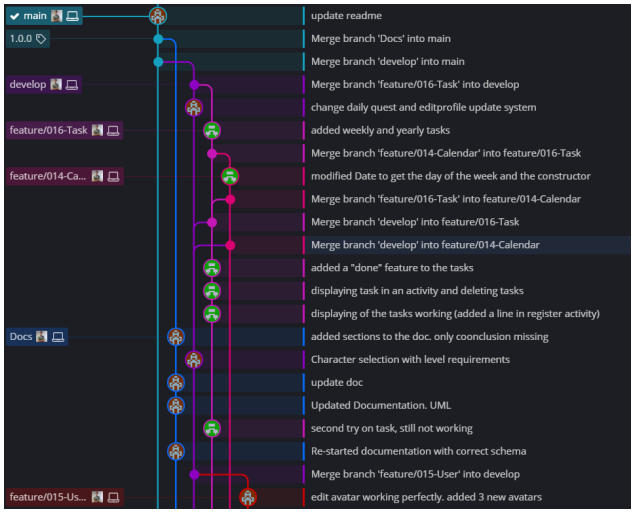

Fig. 27. Gitflow branch strategy being applied

VII. Conclusion

Our goal was to have a minimum viable product of our application, and we succeeded in implementing a complete calendar with tasks that the users can add, delete, and check. Moreover, we have made this app as friendly and kind as possible. We also have implemented levels and customizations. One of the most challenging requirements was to learn how to master Firebase to be able to save the data from the users.

The next main features that should be implemented for the next version of the application should be the overall view of the tasks and the notifications. We did not focus on the overall view of the tasks as we think that it is better to use another language to implement it (like Flutter or React). We really wanted to focus on the basic features of the application, it is to say the calendar and the user features. The notifications have been left over as we needed to implement the task system first. Besides, a feature to improve would be the diversity of the Daily Quests and the customization items.

The team of QuestCalendar is proud of the work we that has been done. Being two members instead of three was a

challenge but we succeed on it, adding a little bit of work in each part. Learn to program on Android was important, to know how apps works from inside and to open new opportunities for our future. The hardest part that was achieved was on sending the data between Fragments and Activities, but we found that saving data on the device or on an external database helps a lot for that.

The team is thinking about continuing the app and launching on some stores, because we think the app is useful and can help people with their daily life by making small challenges that they have to complete and get rewards while using it.

REFERENCES

[1] https://www.figma.com/proto/R38eMVIp0TlyQgalzC26pp/Prototyping-in-Figma?node-id=0%3A1&scaling=scale-down&page-id=0%3A1&starting-point-node-id=0%3A2

[2] https://www.youtube.com/watch?v=8N3vyokTu00

[3] https://0x72.itch.io/pixeldudesmaker

[4] https://github.com/IgnacioNavarro/QuestCalendarApp

[5] https://github.com/IgnacioNavarro/QuestCalendarApp/projects/1