WhatToDo

By
Ünver Can Ünlü
Emre Telli
Özenç Taşdelen

A Dissertation Submitted to the
Faculty of Engineering in Partial Fulfillment of the
Requirements for the Degree of

BACHELOR OF SCIENCE

Department: Computer Engineering

İzmir Institute of Technology
İzmir, Turkey

May 2016

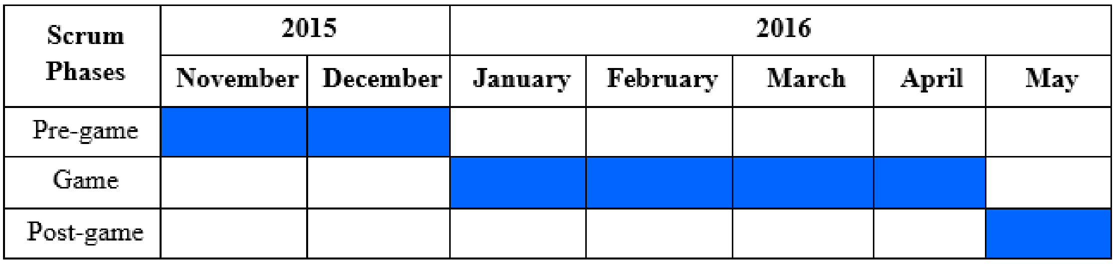
1. THE DESCRIPTION OF WORK

The mobile technology is becoming more popular day by day thanks to developing technology rapidly nowadays. This cause the huge amount of increase in the usage of mobile devices. Nowadays, they affect our daily life directly. Our project's main aim to ease the life of android mobile device users. We focus on lifting effectiveness by developing an android application. And the second aim is trying to avoid forgetting what the important things will be done. Our solution is that we add location and time-based reminder for each to do list with their to-do items for users and taking notes to achieve these goals. User can select the entering or leaving location for their notes and according to this information, they will be notified.

2. THE WORK PLAN

We managed our project using scrum method. Scrum method contains three phases that are pre-game, game, and post-game. Firstly, planning and architecture are important in pre-game phase. Project team structure is defined, software architecture and technologies that will be used are determined and product backlog is prepared in the pre-game. Secondly, game phase is also called development phase. In game phase, sprints are run. Sprints can last 1 to 4 week and consist analysis, design, develop, test and documentation steps. Thirdly, post-game phase is a closer phase and the product is ready to release it. The post-game includes integration, testing, user documentation and marketing.[1][2]

According to our plan, pre-game is done in November and December, game phase is done in January, February, March, and April. We will run four sprints and each one lasts one month. Post-game is done in May.



Scrum Phases	2015		2016				
	November	December	January	February	March	April	May
Pre-game	🟦	🟦					
Game			🟦	🟦	🟦	🟦	
Post-game							🟦

Chart 1 – Gantt chart

The product backlog is shown below:

Product Backlog Item	Estimated Effort
Creating an android application and setting up Android Studio and Android SDK	5
Creating model classes (to-do item, to-do list, note, location, location-based reminder, time-based reminder)	10
Creating view classes (user data, to-do list and note activities with to-do list and user data items adapter classes)	10
Implement SQLite database according to CRUD design.	25
Optimizing database design.	5
Getting debug key for Google Places API and Google Maps API	5
Creating reminder activities (time-based reminder and location-based reminder)	25
Creating reminder controller	8
Creating GPS controller	8
Creating background service	6
Use better main theme to application	8
Creating data transfer activity	12
TOTAL	127

Table 1 – Product Backlog

Sprint 1 Plan:

Before the first sprint, we had already made the mockup. We just started to develop application and we planned that we got an improved mockup after the sprint. Because we planned to use Model-View-Controller (MVC) design, we wanted to create and implement model classes which are to-do item, to-do list, note, location, location-based reminder, time-based reminder. Also, we wanted to create view classes which are user data, to-do list and note activities with to-do list and user data items adapter classes.[3]

Sprint 1 Revise:

We successfully implemented classes and activities as we planned. After the sprint, we have an improved mockup, and this gave ideas about how the workflow of the application's usage to users. The result did not make sense because user data cannot be stored.

Sprint 2 Plan

We basically planned to implement database design in SQLite in Java programming language. Because of MVC, we wanted to write functions as CRUD (Create – Read – Update - Delete) design for each model class object to control the database in controller classes. We wanted to need less database tables to get less database error while using the application and improve performance by optimizing database design.[4]

Sprint 2 Revise

We created all functions as CRUD design. Also, we made SQLite database class singleton design pattern because we did not want to open and close connection to the local database when changing activities or another reason. After this sprint, we got a basic to-do list and note mobile application.

Sprint 3 Plan

We aimed reminder part of the application by creating time based and location-based reminder activities and implementing the works behind them. We wanted to implement how user select location and how location distance is calculated for location-based reminder. In addition to location-based reminder, we desired to carry out time-based reminder operations.

Sprint 3 Revise

We created reminder activities and the application was connected to Google Services. The debug is useable only while the application is been developed. New created reminder activities finished our application user interface.

Sprint 4 Plan

We planned the background works of the application, data transfer feature. For the background works, we planned to create GPS controller, reminder controller and set the sensitive according to requirements. Also, we wanted to define a theme for application and create data transfer activities.

Sprint 4 Revise

We basically finished the application by implementing GPS and reminder controller. We defined an application theme for all activities. This theme used material design as base theme and the theme contained dark and light blue, purple and green colors. In data transfer activity, a good JSON data format was implemented to import and export the use data to the local storage.

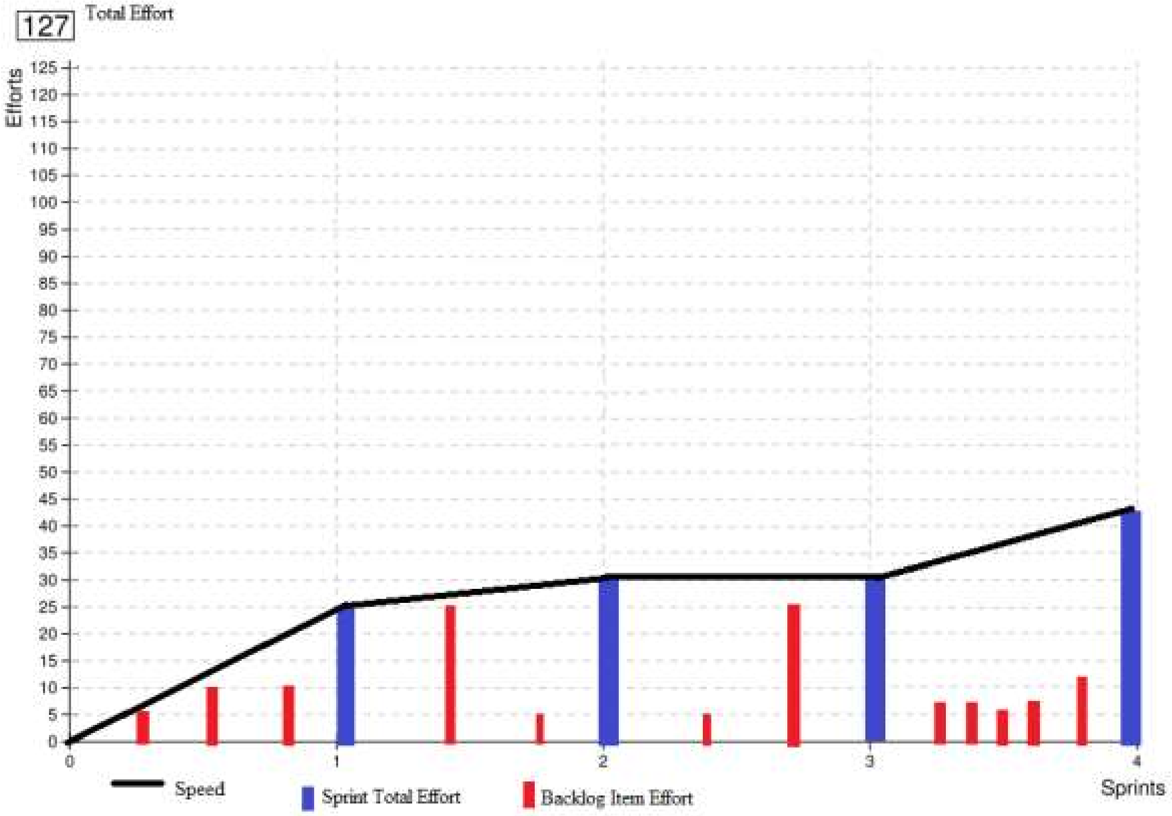


Chart 2 – Burndown Chart

3. ANALYSIS & DESIGN

In the analysis phase, we prepared user stories and themes (functional requirements), non-functional requirements. In the design phase, class diagram, database ERD (Entity Relationship Diagram), domain object model and architecture diagram documents are prepared. [5]

User stories and themes are the functional requirements of the project and these are shown below:

Theme 1: To-do List Stories

- A user can see the list of all to-do lists.
- A user can access an existing to-do list.
- A user can create a to-do list by determining to-do list's header.
- A user can edit an existing to-do list by changing the to-do list's header.
- A user can delete an existing to-do list.

Theme 2: To-do Item Stories

- A user can add a new to-do item to an existing to-do list by determining the to-do item's text.
- A user can edit an existing to-do item by changing the to-do item's text and marking done or undone the to-do item.

- A user can delete an existing to-do item.

Theme 3: Note Stories

- A user can see the list of all notes.
- A user can access an existing note.
- A user can create a note list by determining note’s header and text.
- A user can edit an existing note by changing the note’s header and text
- A user can delete an existing note.

Theme 4: Location Based Reminder Stories

- A user can create a location-based reminder and set it for an existing to-do list by determining the reminder location, the distance to the location and selecting reaching the area or leaving from the area.
- A user can edit an existing location-based reminder by changing the reminder location, the distance to the location and selecting reaching the area or leaving from the area.
- A user can remove the location-based reminder from an existing to-do list.

Theme 5: Time Based Reminder Stories

- A user can create a time-based reminder and set it for an existing to-do list by determining the reminder date and time.
- A user can edit an existing time-based reminder by changing the reminder and time.
- A user can remove the time-based reminder from an existing to-do list.

Theme 6: User Data Transfer Stories

- A user can export the user data to local storage as a new file.
- A user can import the user data from local storage by selecting an existing file.

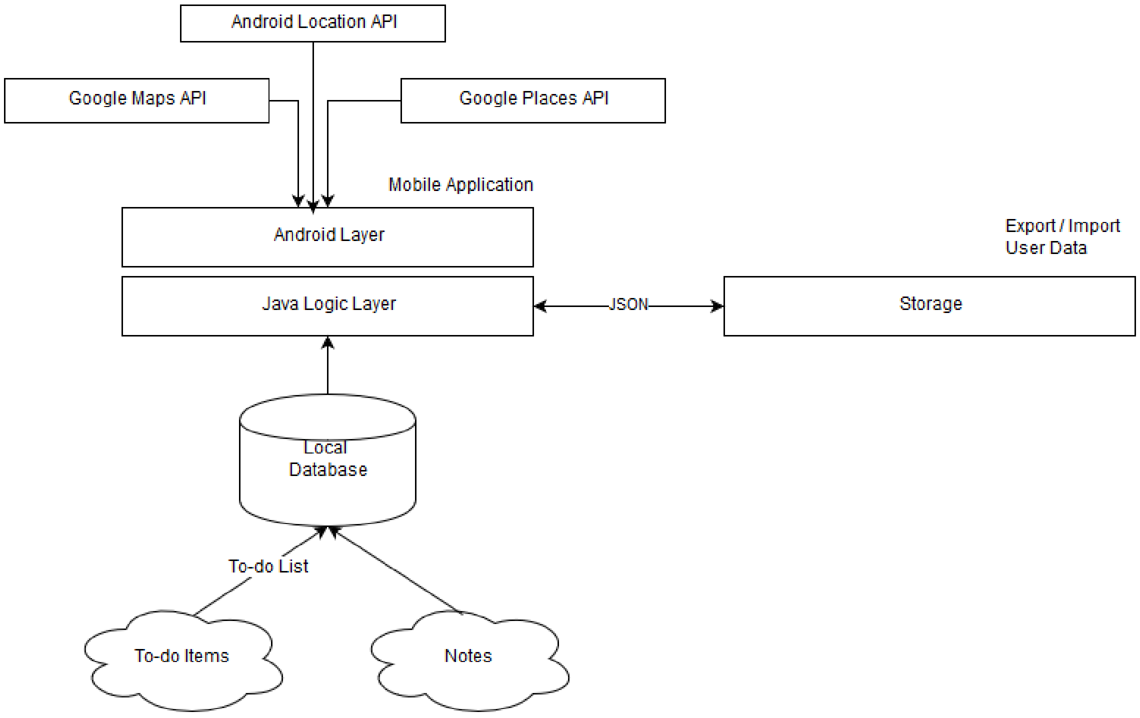
Theme 7: Notification Stories

- A user can be reminded by a notification if required.
- A user can see which to-do list is reminded by clicking the notification.

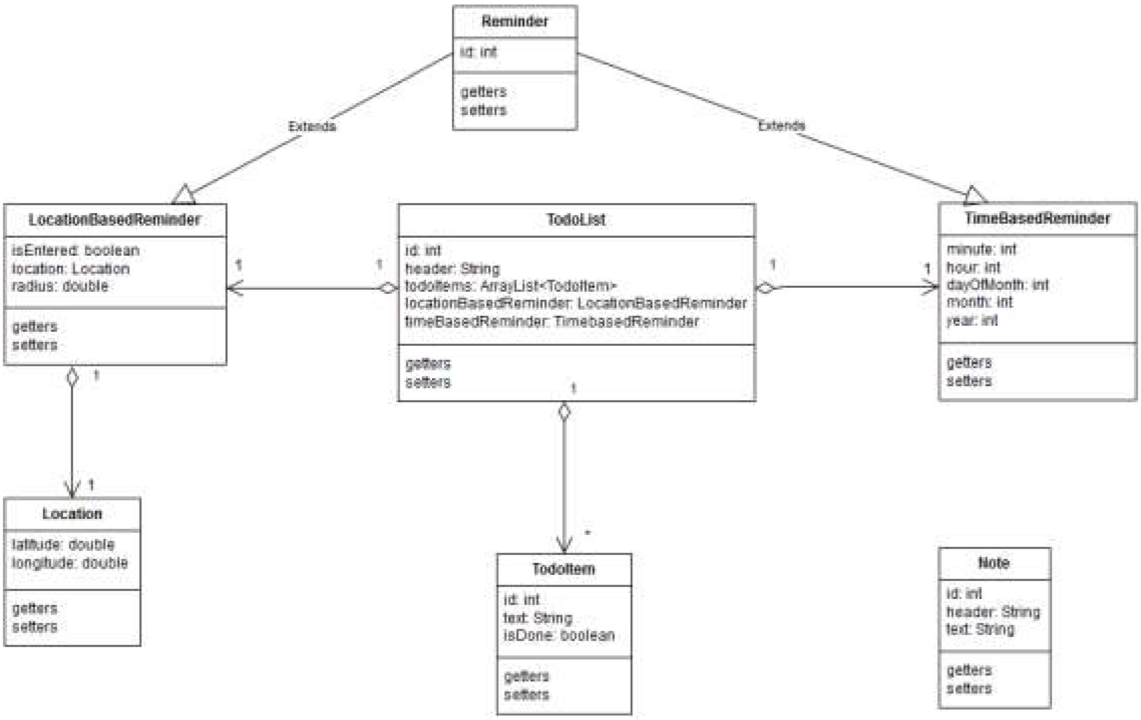
The non-functional requirements are shown below:

- Each reminder is checked each minute.
- Sensitivity of getting the user’s new location is 100 meters.
- User data is stored by using database.
- JSON file format is used for importing and exporting the user data.
- After each user operation, user will be informed by toasts.
- Local time zone is used for time-based reminder.
- If GPS is not active, user will be informed and asked to active GPS.
- A native Android application developed because this application is not web based application.
- This project is implemented in Java programming language because of native Android application.
- Google Places is used to select the location using the location’s name by user.

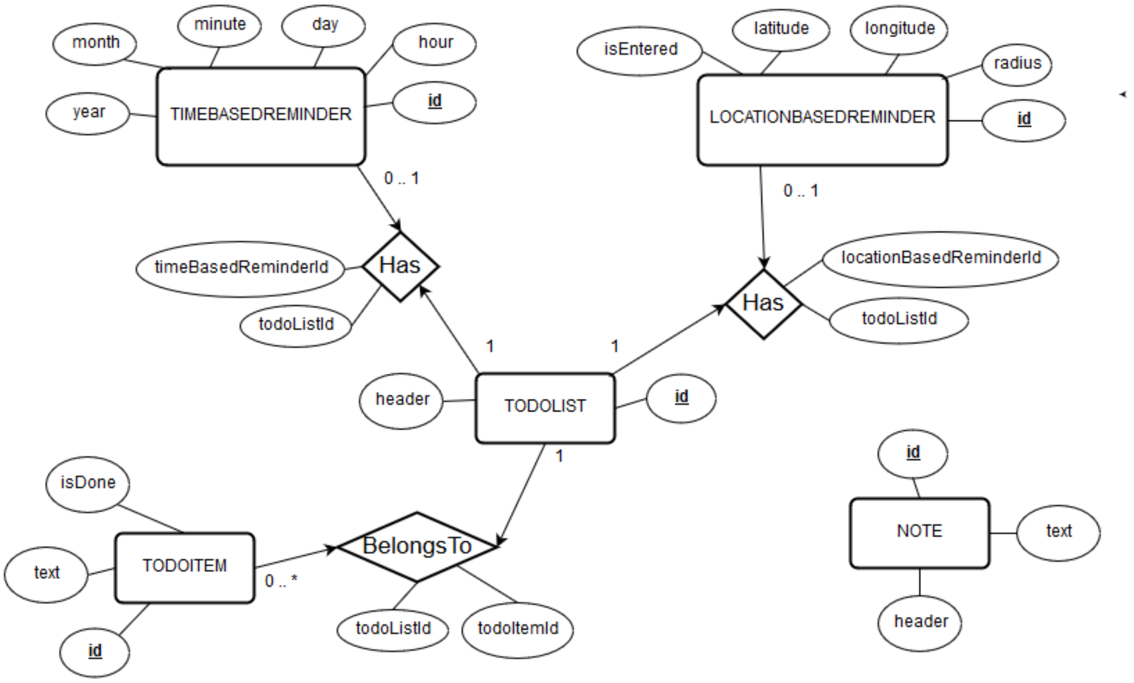
- Google Maps is used to show selected location on map.
- One specific theme which has fixed color in the applications all screens is used.
- Two different icons are used to inform that user data is whether note or to-do list.



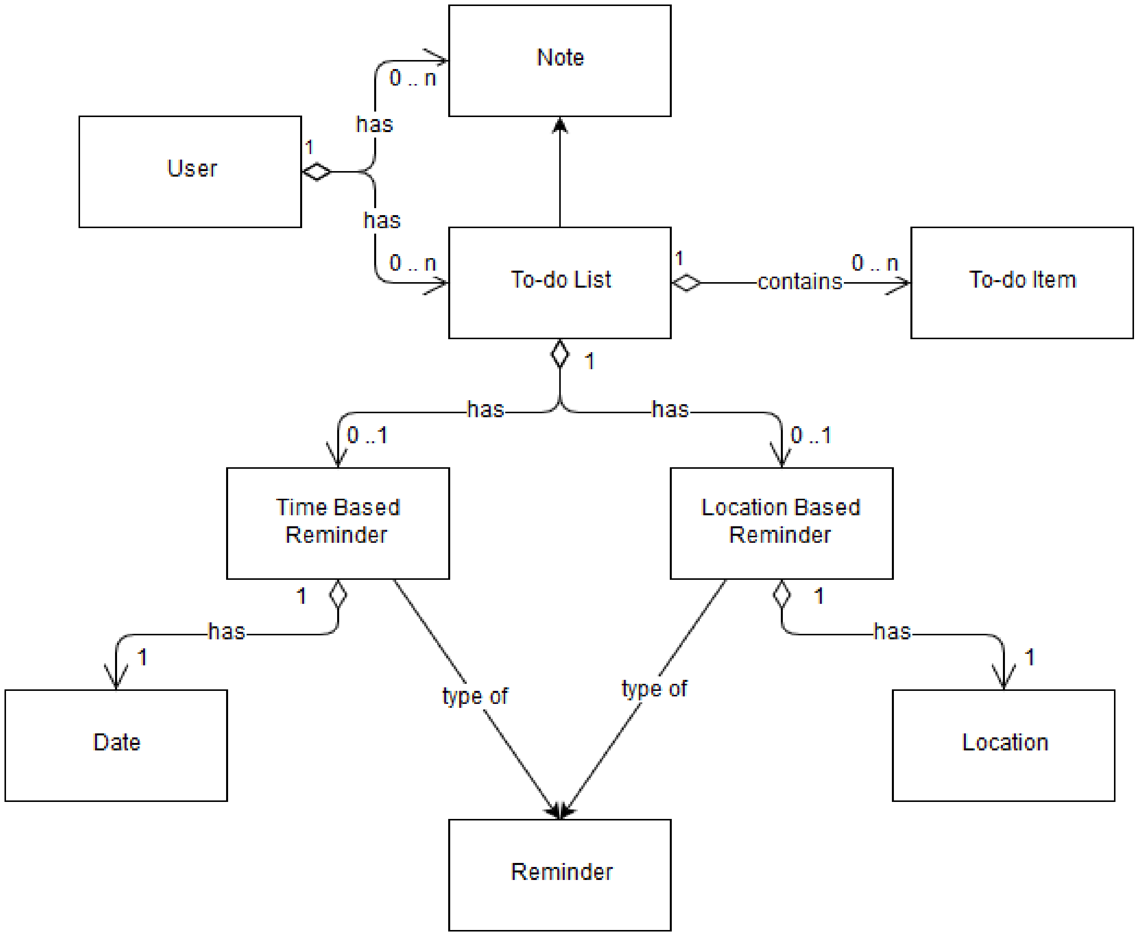
Graphic 1 – Project Architecture



Graphic 2 – Class Diagram



Graphic 3 – Database ERD



Graphic 4 – Domain Object Model

4. SOLUTION/PRODUCT & RESULTS

In the first planned application uses cloud for synchronization when user connected to internet. However, we think that using the cloud is not useful because getting users current location is needed always connected to the internet. Synchronization could be good if the user is not connected to the internet and the last user data is synchronized after connection. We added a button for import and export user data using a file which is JSON format. This makes user can import existing to-do lists and notes if user changes the mobile phone and solve synchronization problem. We planned just use Google Maps API and we recognized that Google Places API more useful because places are already defined for each's geo location. After the changes, the user can create to-do list and note in the application and set reminder for to-do lists. These reminders could be time based or location based. If the reminder is location based, the user can enter radius in kilometer to location and select when the user will be reminded whether entering the zone or leaving from the zone. If the reminder is time based, the user can specify when the user will be reminded. Also, the user can set both of location and time-based reminder. If both are set, location-based reminder will be activated when given time in the time-based reminder passed. Other feature of the application is that the user can export or import the user data. If the user exports the data, all the to-do lists, and notes will be stored as a file. If the user wants to import the existing data, the user must select the previous backup file.

5. RELATED WORK/SIMILAR SOLUTIONS

There are four similar to-do list mobile applications in Google play market. These are Google Keep, Any.do, Wunderlist and Todoist in the most like the least similar order. Firstly, Google Keep's users can create to-do lists and assign a to-do item time or location-based reminder. However, the reminders are not both time and location based although the reminder in our application will be able to both based. Google Keep gives a notification when application when the application's user reaches the given location. Secondly, similarly in Google Keep, to-do lists whose items can be assigned time or location-based reminder can be created in Any.do but only premium users can assign a to-do item location-based reminder. Lastly, there is no option for assigning a to-do item location-based reminder in Wunderlist and Todoist. Users can assign a to-do item only time based reminder. [6][7][8][9]

Similar Application Name	Logo	Time-Based Reminder	Location-Based Reminder	Totally Free
Google Keep	💡	Yes	Yes	Yes
Any.do	✓	Yes	Yes	No
Wunderlist	★	Yes	No	Yes
Todoist		Yes	No	No

Table 2 – Similar Applications and Their Features

6. CONCLUSION & IMPACT

The developing technology all around the world increase the expectation of people. They might be able to want to reach everything with one click or get information which they do not want to keep in their mind about their lives always, with only one notification. For this purpose, to do list applications are developed. To-do lists on mobile phones help people organizing their life and contribute the conservation of environment somehow. For example, basically it avoids the using of papers for taking notes on agendas. After people start to use mobile phone in their almost whole daily life, visibly there is increasing in note taking and to-do list applications for their mobile phones. But they are usually supported by only the time-based reminder. However, time-based reminder for a to-do item can be insufficient. At this point, we thought to bring innovation to this application with location-based reminder.

At the present time, people put in a hard day at their works or another places and they might have crucial things to do, of course they don't want to forget these as well as they don't want to keep these in their mind all day, in the light of these information's we planned to facilitate their life and we thought that let each of these to do items belong to a specific location and people never worry about what they need to do! That is, thanks to our application people will be able to get notification according to their items with location based on to do list. Thus, they won't fall behind their social activity or they won't have an argue with their wives or husbands because of forgetting to buy something because they will be reminded when they get closer to a location where they need to do lots of things. All in all, this application will improve daily life efficiency after the user can be reminded location based in this application.

7. REFERENCES

[1] "Scrum." [Online]. Available: https://en.wikipedia.org/wiki/Scrum_%28software_development%29.

[2] "Scrum Phases." [Online]. Available: http://scrummethodology-org.etltools.net/scrum-phases.html. [Accessed: 30-May-2016].

[3] "Model–View–Controller." [Online]. Available: https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller.

[4] "CRUD." [Online]. Available: https://en.wikipedia.org/wiki/Create,_read,_update_and_delete.

[5] "Entity–Relationship Model." [Online]. Available: https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model.

[6] "Google Keep." [Online]. Available: http://www.google.com.tr/keep/.

[7] "Wunderlist." [Online]. Available: https://www.wunderlist.com/.

[8] "Any.do." [Online]. Available: http://www.any.do/.

[9] "Todoist." [Online]. Available: https://todoist.com/.

8. APPENDIX

- Chart 1 – Gantt Chart
- Chart 2 – Burndown Chart
- Table 1 – Product Backlog
- Table 2 – Similar Applications and Their Features
- Graphic 1 – Project Architecture
- Graphic 2 – Class Diagram
- Graphic 3 – Database ERD
- Graphic 4 – Domain Object Model