

J S S Mahavidyapeetha
**JSS Science & Technology University, JSS Technical
Institutions**
Campus, Mysuru – 570 006,
Karnataka



“DAILY TASK MANAGEMENT SYSTEM”

Seminar report submitted in partial fulfillment of curriculum prescribed for the
award of the degree of

**BACHELOR OF ENGINEERING IN
COMPUTER SCIENCE AND ENGINEERING**

BY

GAGAN RAMESH	01JST16CS033
MUBASHEER FARHAN	01JST16CS054
RAJAT SHRIVASTAVA	01JST16CS081
SHAHID NEHAL	01JST16CS094

UNDER THE ABLE GUIDANCE OF PROF

DR. TRISILA DEVI NAGAVI

LECTURER IN DEPT OF COMPUTER SCIENCE AND
ENGINEERING JSS SCIENCE AND TECHNOLOGY UNIVERSITY,
MYSORE

J S S
Mahavidyapetha
JSS Science and Technology
University
Mysuru-570009,
Karnataka,India



DECLARATION

We Gagan, Farhan, Rajat and Shahid students of 5th Semester B.E ,Computer Science & Engineering at JSS Science and Technology University, Mysuru hereby declare that this group activity was carried out by our group and this report was prepared by us as a part of the course work CS530-Software Engineering under the guidance of Prof. Dr. Trisila Devi Nagavi.

Place:

Mysuru

Date:

5/10/2018

Acknowledgment

First, we would like to thank JSS Science and technology for implementing Software Engineering as a subject for our course.

We would also like to thank our Software Engineering Professor Dr. Trisila Devi Nagavi for giving us an opportunity to do this project about task management system.

We would also like to thank Mr. Klevin Mathews for providing us with the requirements necessary for implementing this project

We would also like to thank our friends who were our backbone and helped us in collecting the requirements for our project.

At last we would like to thank all the other people who were involved directly or indirectly in collecting the requirements.

- Gagan R
- M Farhan
- Rajat S
- Shahid N

Abstract

This document serves as a requirements engineering study for our academic project Daily Task Manager. Each project requires research into the features, functions and design that the developers must implement to ensure user satisfaction. This document contains the direct responses from potential users, the analysis of their requirements and a comprehensive list of all features and models that must be designed from aforementioned analysis. User interactions, system interfaces and system specifications are all recorded in this document along with validation and requirement management. The goal of this report is to provide the designers of the product with clear specifications and a clear concept of the implementation of the system. The structure of this document follows the standard IEEE template.[3][2]

Contents

1	Inception	1
1.1	Understanding the problem	1
1.2	People that want a solution	1
1.3	Nature of the solution	1
1.4	Communication and collaboration	2
2	Elicitation	4
2.1	Questionnaires	4
2.1.1	Businesses And Stores	4
2.1.2	Students	5
2.1.3	Travel Enthusiasts	7
2.2	Association of requirements	8
3	Elaboration	9
3.1	Primary Block Structure	9
3.1.1	Database System	9
3.1.2	System Processes	10
3.1.3	Application Interface	10
3.1.4	Users	10
3.2	Secondary Block Structure	10
4	Negotiation	12
4.1	Features	12
4.2	Result	12
5	Specifications	14
5.1	Introduction	14
5.1.1	Purpose	14
5.1.2	Scope	14
5.1.3	Overview	14
5.2	Product Perspective	15
5.2.1	Software Interfaces	15
5.2.2	Hardware Interfaces	15
5.2.3	Communication Interfaces	15
5.2.4	Memory Constraints	15
5.2.5	Operations	16
5.2.6	Site Adaptations	16
5.3	Product Functions	16
5.4	User characteristics	16
5.5	Constraints	17
5.6	Assumptions and dependencies	17
5.7	Specific Requirements	17
5.7.1	Performance Requirements	17

5.7.2	Logical Database Requirements	18
5.8	Design Constraints	18
5.8.1	Security	18
6	Validation	19
6.1	Validation Checklist	19
7	Requirements Management	21
7.1	Management and Trace Matrices	21
7.2	Requirements List	22
8	Gallery	23
8.1	Entrepreneurs	23
8.2	Students and Travel Enthusiasts	24
8.3	Group Meetings	25

The aim of the daily task manager system is to keep track of user activities, provide timely notifications to users and help increase productivity and work efficiency by providing overall task statistics. The management system will also help improve efficiency for collaborative group tasks, routines and calendar-synced activities.

1.1 Understanding the problem

As students we often find ourselves struggling to stay on top of our work load, keep all our deadlines, maintain an effective schedule and manage our personal and social lives. Similar to us, there many other people from various fields and different walks of life who experience the same problem of time management. Due to this, we sometimes miss or forget our priorities leading to more unnecessary stress. There are several scenarios where a person might need to keep track of their tasks and prioritise.

Keeping track of all the important tasks of your day is not an easy job and requires considerable amount of planning. Often in the way we plan out our jobs, one task is related to another and hence every single task becomes affiliated. The failure to complete a single task could result in the failure of the entire task list. This can create problems with deadlines and other time sensitive submissions.

Another problem is that not everyone's priorities and activities are the same. A group of people trying to complete a project or any other kind of work should always be on the same page and have a similar mindset about their situation and work flow. Not having a clear understanding of all the remaining tasks and their importance can cause more issues.

1.2 People that want a solution

The people who want a solution consist largely of students and small business owners. However due to the general nature of the problem at hand, our solution can be applicable to anyone who wants to manage their time. This could range from travel enthusiasts and event planners to teachers and other trained professionals. It is clear that they want a system that is reliable, portable and most importantly helps them improve general work flow performance.

1.3 Nature of the solution

The nature of the solution designed and implemented by our system will follow the following guidelines:

- Primary purpose should always be to keep the user's mind at ease, knowing that the management will track their actions reliably

- The system must never skip or forget an urgent task
- Should ultimately lead to increased productivity and satisfaction

1.4 Communication and collaboration

Due to the extensive nature of the problem that this system overcomes, anyone with a schedule or prioritised tasks is a potential client or stakeholder. To obtain a more specific and realistic approach towards obtaining requirements we reduced our target audience to three specific clients:

- **Students**
The problems faced by students was the main source of inspiration to solve the issue of time and activity management. Naturally we can deduce that professors and other lecturers could also benefit from our system due to the large work loads that they have.
- **Small businesses like cafés and stores**
Most restaurants and shops have a fixed routine that they follow everyday before opening and after closing. Their activities generally follow a fixed routine or pattern and change very rarely based on specific situations and include tasks for preparation for the next day.
- **Travel Enthusiasts**
Before leaving for trips, whether long or short, planned or unplanned, there is always some amount of preparation required. Travellers have a set of activities that they need to be in order before they go for trips eg. bill payments, medicines and supply purchases and other essential requirements for their trip.

Our team was in communication with all three clients and prepared a set of general questions for each of these groups along with some specific questions for each set. These questions were structured in order to obtain the maximum possible information from the users. The elicitation (section 2) contains more detailed interviews with clients.

During our first meeting, Farhan came up with the idea for managing projects and activities for students on a daily basis, after which we added features and decided to make a more general task management system that not only students, but other people could use as well. He also came up with ideas for more potential user target audiences. These are discussed in section 1

The next two meetings were crucial in which we had discussions with the clients. We used the questionnaire to obtain all the possible requirements from them. The questions were designed by Rajat and Gagan, the requirements analysis and interview itself was conducted by Gagan. In future meetings it was decided that we needed more information and more interviews were conducted by all the members of the team, as discussed in section 2

Finally in our last meeting, Shahid produced a list of features and functionalities along with their dependencies after analysing the requirements carefully in section 5

Rajat and Shahid were responsible for the interface and system models. After taking into consideration all the requirements and information gathered from the clients, we have constructed this report for requirements engineering on our application.

The photos taken during client meetings and group discussions have been added to this document in section 8

The following questions were structured to obtain all the required information from the target audience. It is to be noted that this set of questions are standard and were answered and verified by each customer. The following section contains the questions along with the paraphrased responses of the user.

2.1 Questionnaires

2.1.1 Businesses And Stores

This set of questions was answered by the owner and manager of Rhythm N Brews, Klevin Matthews. The café specialises in high end coffee and other continental delicacies. His requirements can be taken as the standard for small businesses and stores that follow a similar work flow.

1. Do you currently own or use a system that aids you in keeping track of your extensive activities or tasks? If no, then what is your process?

We currently do not have any system in place to keep track of daily activities. I have used other workbench style applications before but they were difficult to use and didn't run very well on our computers. Originally, the systems we used were to keep track of prep time before and after closing time and maintenance of the café itself. Right now we operate everything manually, by practice and memory. My manager and I handle most of the important tasks that involve money, and oversee the chefs and other employees to ensure that they complete their work.

2. What features would you consider essential in a task management system? Are there any in particular that you would use?

A task system that I would consider useful should have a feature that let's me monitor all my employees work pre-opening and post-closing without actually having to be here. Our previous workbench system only allowed us to monitor tasks from the café itself. I would prefer a mobile application to that. Another feature I would like is the option to add or remove tasks to an already set routine. We have a set routine here for most days, but our workload changes on weekends or other holidays for which we have a different routine.

3. Do you notice any flaws in your current management system? Is there a way for us to eliminate this problem?

Our current system is not automated, it is just by memory and practice so of course sometimes there are mistakes. If I can create a list of tasks to complete everyday and monitor it then that would definitely be helpful. As for flaws, in our previous systems there was no mobility, and could be used only by one person at a time. The system we had was designed more towards running the kitchen than the entire establishment. I think that having a more general task manager type of system would be useful for us.

4. *If our application was available to you right now, how would you use its features? Does the application help in simplifying tasks?*

If it is possible to set a routine on your application along with regular reminders, I would use the system to keep track of tasks that the workers forget or lose track of easily. Off the top of my mind I can say: to check wash room cleanliness, to check for tissues and soap in the wash room, to check if all appliances in the kitchen are off before leaving and to check if the caretaker watered the plants outside. These things are very easy to forget on a daily basis and if ignored for a long time can have consequences on the business itself. Naturally, having a notification system to check these tasks would be helpful.

5. *Can you think of any reasons why our product would fail to deliver on performance or otherwise?*

I have no specific idea about this. If the system doesn't log the tasks properly then some tasks might get skipped or my employees might end up repeating tasks that were already complete. Another issue would be that my routine isn't usually fixed for the weekends so if the management system can't update to new routines then it would fail to perform.

6. *Is there anyone else who could be considered a stakeholder in this product that we should talk to?*

In this establishment there is no one else. If you are considering going to bigger restaurants, then you might have to add some more features that include billing options, daily revenue, the amount of petty cash used and other financial and stock details about the kitchen.

2.1.2 Students

This set of questions was answered by multiple students in and around JSS Technical University. Their answers have been aggregated and shown below, since a majority of the students had similar answers. The candidates chosen for the questionnaire were all between the ages of 19 to 22 ie. PU and university students.

1. *Do you currently own or use a system that aids you in keeping track of your extensive activities or tasks? If no, then what is your process?*

We keep track of activities mostly by memory, although sometimes when we have too many deadlines, we note them down on paper or on our phones. Our schedules are not fixed and generally very volatile. We have tests during the end of the month and most of the projects are during the end of the semester, so we don't have to worry about keeping track of these tasks for the first one or two months. Eventually when the deadlines come around we start working frantically and it's always a struggle.

2. *What features would you consider essential in a task management system? Are there any in particular that you would use?*

Features to remind us to do the tasks diligently instead of wasting time on our phones. A regular notification system would be really useful. We also have group projects very regularly so having the ability to have groups and group task will make coordination simpler. A group chat and an administrator to keep all the members in check to make sure everyone is doing their tasks properly will also help us a lot.

3. Do you notice any flaws in your current management system? Is there a way for us to eliminate this problem?

Our current system is that we have no system. It all depends on the amount of work we get and the pressure of the deadlines. We(some of us) have used 'study tracking apps' before but they were of no use and lacked the features to coordinate with other people. We want something consistent that tracks tasks for us over a stretch of time, so that it doesn't become overly difficult towards the end of the semester. We need a more organised structure to approach these problems.

4. If our application was available to you right now, how would you use its features? Does the application help in simplifying tasks?

Our main objective would be group discussions and group task lists. The ability to maintain a set of tasks specific to each person on their smartphones would make it easier to progress. Our productivity would definitely go up and we won't have to remember as many things. Daily tasks would be good for general activities, and don't necessarily have to be related to college. Maintaining a priority list would also help us a lot during fest events and other organised club events because we always have so much work during those times. Having our own separate lists for college clubs would save us a lot of trouble.

5. Can you think of any reasons why our product would fail to deliver on performance or otherwise?

If the notifications are not regular enough then no one will remember to use the task management system, but if there are too many notifications then people will get irritated and uninstall the application so it needs to be very balanced. Also if the system doesn't provide statistics about our progress then it will feel like we aren't getting better and that the application is not helping. There needs to be a way to make the user feel validated.

6. Is there anyone else who could be considered a stakeholder in this product that we should talk to?

Teachers, professors and other lecturers have lengthy lesson plans to make, question papers, solution keys, syllabus details and not to mention for multiple classes. A system for them to manage all these activities could be useful, especially if they could carry it around on their phone.

This set of questions was answered by a group of young adults who are travel enthusiasts and go on trips to remote places fairly often. They prefer road trips over other forms of travel and have been doing this for the last 4 years. Their responses have been aggregated due to similarity between individual answers.

1. *Do you currently own or use a system that aids you in keeping track of your extensive activities or tasks? If no, then what is your process?*

In general we don't keep a task manager system, we have no use for it. However before our trips one of us usually writes down all the things that we need to prepare for the trip to go smoothly. Now over time most of these things have become part of our process just by practice, and we remember most of the things on the list without having to write it down. But sometimes there are still cases when someone forgets to pack batteries for the torches, or forgets to check if the car batteries are charged.

2. *What features would you consider essential in a task management system? Are there any in particular that you would use?*

For us specifically we need an urgent task list because our needs are time bound. We don't usually plan our trips very far in advance, maybe a day or two in advance. So in the span of two days we have a lot of work to do, including transport arrangements, accommodation, activity planning and buying supplies. These things are very important and if we forget to do this then the entire trip becomes difficult. Another feature that would help is keeping track of expenses as we travel, helping us stay within our financial range.

3. *Do you notice any flaws on your current management system? Is there a way for us to eliminate this problem?*

The biggest flaw is not having a proper reminder system and not being able to keep the urgent and casual tasks separately. Even when we manage to note down all the tasks we need to complete before a trip, the order in which we plan the events makes a big difference in our process. If we leave the urgent tasks right up to the end and it gets delayed for some unforeseeable reason, then we usually have to cancel the trip.

4. *If our application was available to you right now, how would you use its features? Does the application help in simplifying tasks?*

One of the main things we would use the application for is to make sure we are prepared for the worst situations. From experience we can say that we've had a bad time because we forgot medicines, batteries or other supplies that are required on all our trips. We generally prefer travelling to locations that are a little isolated and forgetting tasks like this can create major problems. Having coordination when planning trips like this is extremely important.

5. *Can you think of any reasons why our product would fail to deliver on performance or otherwise?*

The idea of this application is pretty straightforward so there aren't many cases in which it can go wrong if it is implemented correctly. The only case we can think of is the group lists aren't managed properly then it can cause repetition of tasks. Another issue is that the places we visit during our travels are usually remote places and don't have reliable internet connections, in which case the group lists wouldn't work at all.

6. *Is there anyone else who could be considered a stakeholder in this product that we should talk to?*

Maybe postal services, courier services or even regular deliverymen could benefit from this system. They usually have tasks that need daily updating and are of high priority. A system from them to keep track of tasks could help them increase efficiency.

2.2 Association of requirements

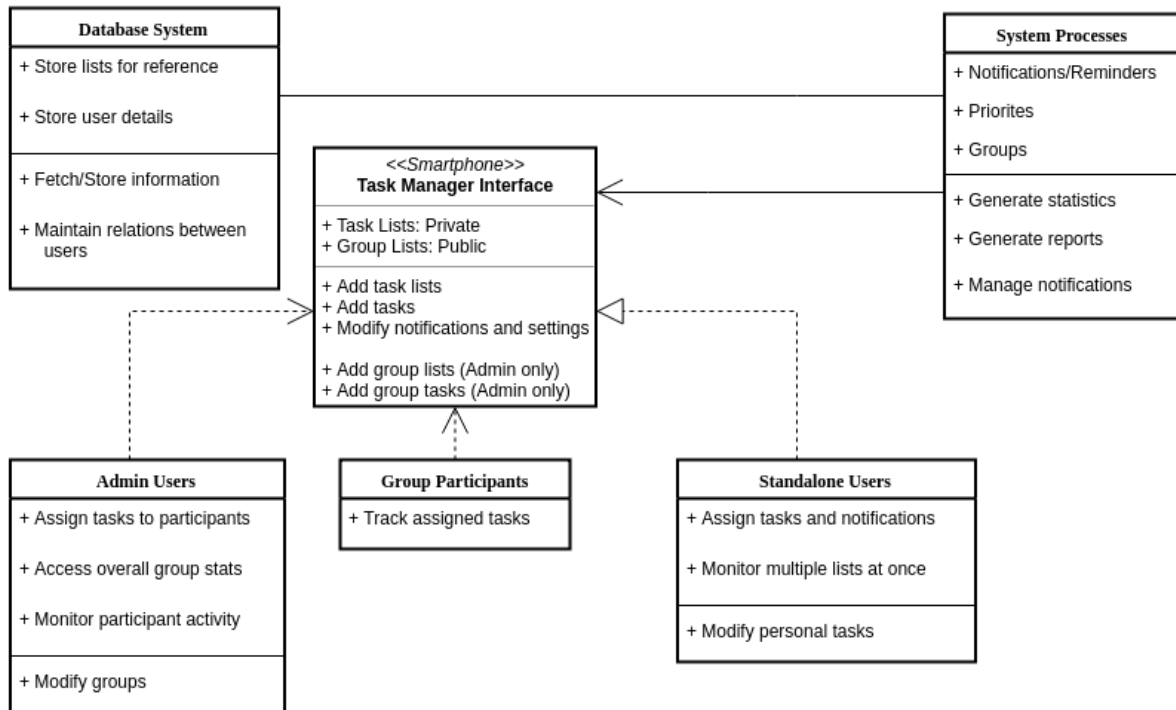
From the above questionnaire we have collected all the necessary requirements and created a list to attribute all the features to their source.

- Students: list and task creation, regular notifications, group lists and tasks, task and user statistics
- Entrepreneurs: list and task creation, customised notifications, group lists and tasks, routine check lists, administrative control
- Travel Enthusiasts: list and task creation, customised notifications, group lists and tasks, prioritised and urgent tasks, task and user statistics

These features that are attributed to their sources can be used to correctly construct the trace matrix (as discussed in section 7.2) and have validation of requirements as shown in section 6).

From the information collected via the questions in the previous sections, we were able to create a refined model containing the basic functions that the system will implement, the interactions between the users and the task management system as well as the overall association between each module.

3.1 Primary Block Structure



The following subsections define the primary structure of the task management system. Each block represents a module of the system. The top half consists of the features available to the module, and the bottom half consists of the functions that the module will use to implement these features.

3.1.1 Database System

The database system is responsible for handling storage of all the tasks and their respective task lists, along with user statistics and their history. It will be hosted online to ensure that users have access to all their previous data from any device. A third party service will be responsible for handling security and integrity of the database. It will also be responsible for keeping track of all user relations within a group structure.[1]

This module documents the features and methods implemented by the core processing unit of the application. The system is responsible for handling the following procedures:

- notifications/reminders for every task list and customised reminders for each task
- handling of prioritised tasks and calculation of efficient task orders
- group construction and management for multiple users
- generating suitable reports along with statistics to track efficiency

These procedures will be designed and implemented with abstraction, ie. the working of will not be visible or available to the users and they will be aware only of overall functionality.

3.1.3 Application Interface

The target hardware interface of this system is smartphones and other smart hand-held devices that support Android operating system. The interface design should be minimal with distinct options for each feature with clearly stated instructions. The core functionality should give the user a simple way to create a task list and check them away as they complete the task. The more complex features of group lists and administration should be available only to the owner or administrator of the list.

All interface options will follow the Android UX-B1 and UX-S1 standards.[1]

3.1.4 Users

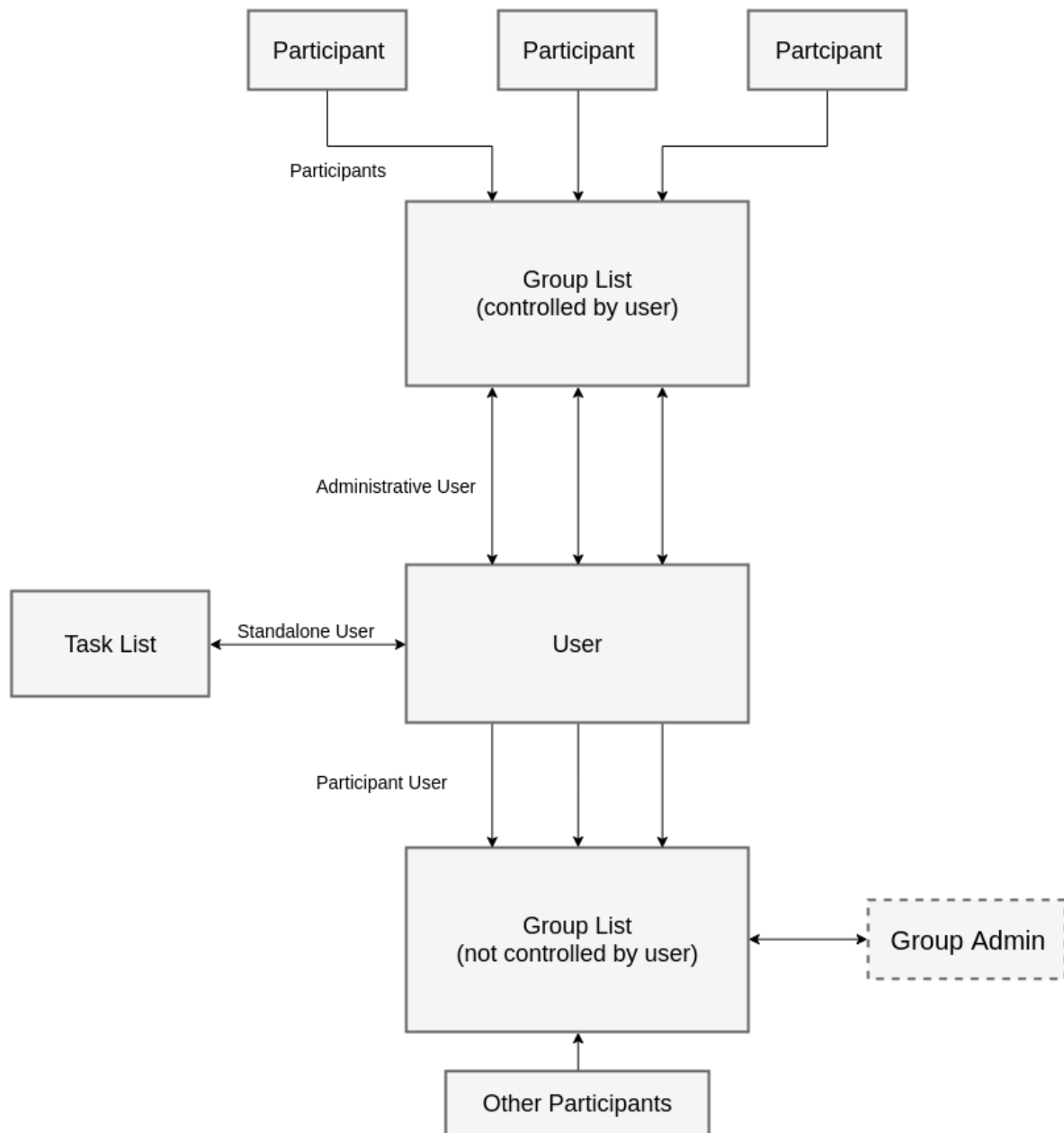
There are three categories of users for this system however they are not exhaustive ie. one person can belong to any number of categories stated below:

- Administrators
- Participants
- Standalone users

A person using will belong in at least one of the above categories, and can be part of all of them simultaneously as well. The role of the user is defined by their interaction with other users and the system. As a result, the application must provide the user with the functionalities that each category requires.

3.2 Secondary Block Structure

Standalone users will have lists containing activities that are purely their own ie. tasks can be completed only by them and don't regard any other person or user.



Administrators will have group lists containing activities regarding multiple people and can control the tasks of each individual person in the group list. In addition to this the administrator also controls notifications and priorities of each task in the list.

Participants are users who are assigned tasks in group lists. They cannot choose or update their task without administrator privileges.

It is clear from these block diagrams that a single user can fall under all three categories at the same time and hence the system must be able manage and provide distinct features to the user for each categories function. The administrator of one group could be a participant of another group, and maintain a standalone list of their own simultaneously.

It is not unusual for customers and users to ask for more than can be achieved, given that the developers may have several problems to be dealt with. It is also relatively common for different customers or users to propose conflicting requirements, arguing that their version is essential specifically to them.

Inception phase made it possible for us to think of an application such as the Daily Task Manager. This concept was further elicited and elaborated to have a clear understanding of the various stakeholders, all the various requirements that the stakeholders demanded, and also ideas on how the various technical implementations could be done in the next steps of software development.

Our team also had in mind the deadline of the requirement sheet submission and thus had to cut down a few requirements that were out of scope of the project and that were out of scope of the available skill set that our team possessed. Our development team had to keep in mind that we were developing a task management system that was not specific to a single type of client but to general users who wanted to increase their productivity. As such, we had to negotiate with clients who had system specific requirements that would benefit them directly, but not prove to be useful for other types of clients.

A few examples of such requirements can be cited as follows:

4.1 Features

1. The requirement of having a group chat feature was requested by multiple clients along with the group lists feature. This requirement cannot be implemented by the available skill set that our team possesses. Due to the tight schedule and deadlines of the development team, we had to discard this requirement even though it was possible for the team members to work off the required skill set and get themselves trained in order to implement the above cited requirement.
2. Another requirement which was actually out of scope of what our project is actually meant for, was brought forward by Mr. Klevin, the owner of ‘Rhythm N Brews’, a small café whose business could use a management system. The requirement of Mr. Klevin was to have a feature that would make it possible for the users to view the billing information on the application directly and also to make payments from it. Since our application’s main focus is to have a management system that can manage all the tasks entered by the user, this requirement seemed out of context and hence we negotiated with Mr. Klevin and discarded the requirement.

4.2 Result

Our team decided to negotiate such requirements with the stakeholders so that we could strive for a “win-win” result. In this “win-win” situation, what we intended for was that the stakeholders would win by getting all the functional requirements which were absolutely

necessary and those which would satisfy their needs, and also we would win by working to realistic and achievable deadlines.

Thus, during the negotiation phase, our team successfully identified the key stakeholders and determined the “win” conditions of the stakeholders. Then we approached the stakeholders and negotiated the stakeholder’s “win” conditions in order to have them satisfied with those requirements and also to make us complete the task within the specified deadline. This was a key step for our team to carry out the further plans which include the actual design and implementation of the project.

The specifications of the Daily Task Manager are shown in the following sections as a Software Requirement Specifications document.[3]

5.1 Introduction

The following subsections of the Software Requirements Specifications (SRS) document provide an overview of the entire SRS. This document provides details on how the developers will design the system, however this document is not for designing.

5.1.1 Purpose

The purpose of this document is to present a detailed description of the Daily Task Manager application. It will explain the purpose and features of the application, the interfaces of the system, what the application will do, the constraints under which it must operate and how the application will react to external stimuli.

5.1.2 Scope

This software application called the Daily Task Manager will be a system targeted towards students and event planners. The system will be designed to maximize the user's productivity by providing tools to assist in managing day to day tasks, collaborating with other people over planned events and tracking progress over the course of multiple days.

By maximizing the user's work efficiency and providing detailed statistics of the user's work flow the system will meet the user's requirements while remaining easy to understand and use.

More specifically, this application is designed to allow the user to allocate tasks on a daily basis and keep track of their progress as the day goes by. The tasks can range from any mundane activity to urgent issues as the application will support the option of assigning priority. The application will facilitate communication between multiple users to keep track of tasks that require collaborative work (eg. group projects, events, travel). Finally the system will provide detailed statistics about completed tasks, productivity charts and efficiency percentage for each day the user assigns activities. This will aid the user in achieving their goal of increased productivity and better work flow.

5.1.3 Overview

The next section of this document gives an overview of the functionality of the application. It describes the informal requirements and is used to establish a context for the technical requirements specification (section 5.7). This document is written primarily for the developers and describes in technical terms the details of the functionality of

the product. Both sections of the document describe the same software product in its entirety, but are intended for different audiences and thus use different language.

5.2 Product Perspective

This system is independent and self-contained and is not a part of any larger system. The application will operate on every individual users Android device and provide all the mentioned features. The developed system is comparable to other 'to-do list' category applications in the market, with the exception that this system allows group lists and prioritised tasks.

5.2.1 Software Interfaces

The target device for this system is a smartphone which runs Android operating system on it. As a result, the interface will consist of the standard buttons and menus provided by Android and will adhere to UX-S1 Android standard. The standard dictates a set of API's that can be used to ensure quality and uniformity across all devices. [1]

The user can communicate with the system through the touch screen only, no other input option is available. An interactive and user friendly GUI should guide the users to their desired function. The application will have access to the screen for display, and the device speakers to provide notifications for the tasks. The system will not have access to any other output ports.

5.2.2 Hardware Interfaces

The system must strictly run smartphones. There are no other hardware interface requirements other than this.[1]

5.2.3 Communication Interfaces

The application will use standard HTTP protocol for communication between users during collaboration for the group lists, and for fetching and storing of results during statistics calculation.

5.2.4 Memory Constraints

The application will occupy a maximum of 25 megabytes of memory during installation. No caching operations are available.

The system contains only one mode of operation which is common to all users. A third party hosting service provides us with backup and recovery options so that the users can access their data across multiple devices. Google will be responsible for handling the user accounts, as well as their data backup.[1]

5.2.6 Site Adaptations

The application will be run and maintained by Google servers on the Google Play Store. There are no specific functions required by our system prior to site activation.[1]

5.3 Product Functions

The features provided by the function can be summarised as follows:

1. Creation of tasks list and monitoring the same
2. Assigning priorities to the tasks
3. Ordering the tasks
4. Setting custom reminders for notifications throughout the day.
5. Creation of group task lists with an administrator and multiple group members
6. Moderating the group list from anywhere, administrator can assign tasks to each member
7. Creation of a routine tasks list that can be kept constant.
8. Statistics for overall task progress for the current day
9. Reminder to set the tasks for the next day.

A diagrammatic representation of the relation and communication between these operations has been specified earlier in sections 3.1 and 3.2

5.4 User characteristics

The intended users of our project have been categorized into 3 groups:

- Students
- Travel Enthusiasts
- Entrepreneurs

However our project is applicable to anyone who has a list of tasks to be managed. The users of our project are assumed to have knowledge of how to use an android phone. Not

much of technical expertise is required for the users to be able to use our application. The main idea of how this project was brought about was due to the fact that many users falling in various categories especially Students, are exposed to a lot of tasks to be done every day and this characteristic of students is what made us come up with this project. We have kept in mind the generality of user base and hence will try to implement the project in as much simple way as possible.

5.5 Constraints

The major constraints that limit our project's options include the following:

- 'Difficult to implement' requirements which include a feature like group chatting
- 'Higher order language requirements' like to have an option where a user could view the billing and other things.

The above are just a few non-functional requirements given in the customer's language.

5.6 Assumptions and dependencies

Our team also thought of the various factors on which our application depends. For example, our project is an android-based application, and it would fail if a user (of any category) tries to install the application on a device that does not use an Android operating system. Our application would fail in case of windows and iphone users. This application is being developed for android of minimum version Gingerbread. Any android user having a version prior to Gingerbread would not be able to install this application.

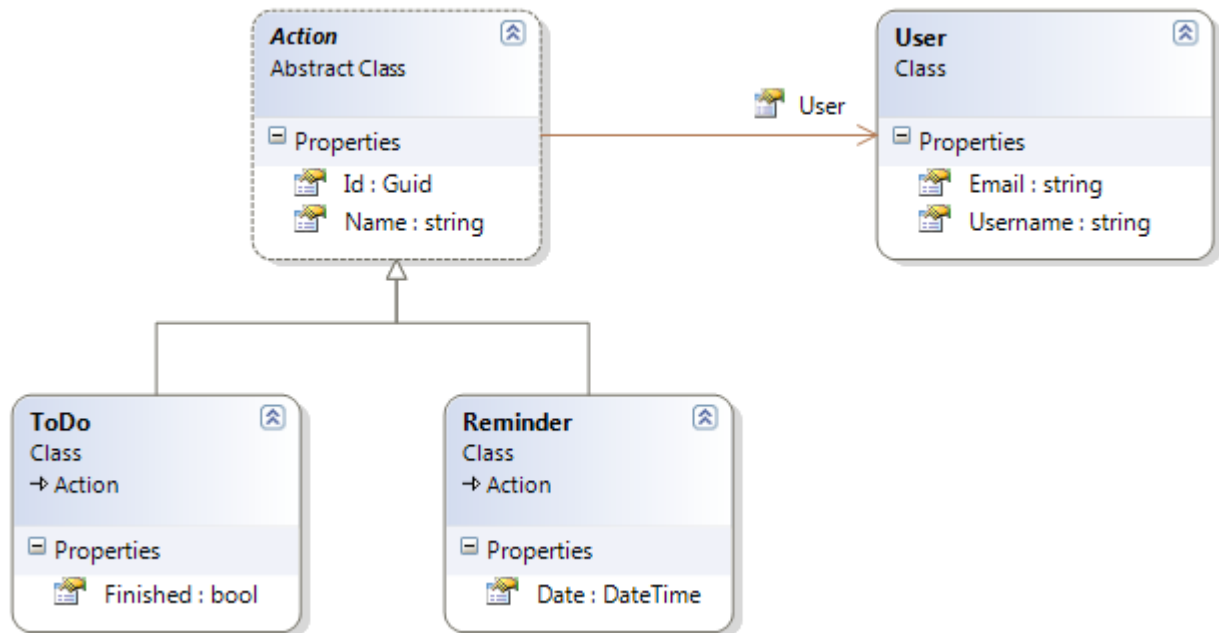
5.7 Specific Requirements

This section will explore the various cases the user will experience during the operation of the system and the implementation of the functions discussed earlier in section 5.3

5.7.1 Performance Requirements

The performance of the system is required to be dynamic as the interactions with the system change with the number of users. The number of people taking part in group lists can also change based on administrator activities. The system shall handle up to 10,000 users simultaneously and will handle each of their information solely in text/date and number formats.

The types of data stored in the database will consist of text, dates and numbers only. Each of these data fields will be used very frequently as it is the basis of the design of this application. The integrity constraints will be implemented by the Google Play Services to verify that each user has access to the application only through one device at a time.[1]



The ER model above shows the basic relational connections between the user, the tasks, the actions that the user and system can take as well as the reminder system. A more rudimentary structure of the system is represented in section 3.1

5.8 Design Constraints

5.8.1 Security

The application deals with privacy sensitive information namely the users schedule and their activity over the a long period of time. The security of each users data and their data will be provided by the Google Play Service that will host the application. The communication that takes place between group participants will be encrypted, however they will be persisted on the server for future calculations and statistics.[1]

In addition to this, the system will maintain a change log to keep track of all agents that access private information. This will help keep sensitive data in an organised manner.

Validation of the collected requirements plays an important role in the overall process of requirements gathering. The process of validation of requirements involves activities like checking for any ambiguities in the requirements collected, omissions of any requirements and finding out if any inconsistent requirements are present or not. Once the requirement model is shown to the stakeholders, the stakeholders prioritize the requirements based on their needs and we, as requirement engineers, group the requirements into packages so that they can be implemented easily as software increments[2]. During this step, the requirement engineer must ask questions that are shown in the following subsection.

6.1 Validation Checklist

1. *Is each requirement bounded and unambiguous?*

We went through the requirements which were mentioned previously and found out that there were a few ambiguous ones, like the stakeholder category of Students gave us a lot of ambiguous requirements eg. some students requested for regular notifications to complete tasks through out the day while some requested for sparse notifications as the find continuous reminders irritating. We eliminated such requirements by introducing the option to provide custom notification options for each task list.

Our team provided abstract details to the stakeholders which hid the implementation part of the project. This made easy for the team to gather requirements without confusing the stakeholders of the implementation part. A few requirements which were out of scope of the project were realized in this stage, and were eliminated. The students category of stakeholders requested for a feature where each of the users of the application could communicate with each of the other user through a feature like 'Group Chat'. This requirement was out of scope of our project and was not feasible in terms of implementation, so our team eliminated this specific requirement. Apart from this, the entrepreneurs category of stakeholders wanted to have a feature where the users could have an option to view the bills in the application. This requirement was out of scope of our project, since the main goal of our project is to have a managing system which can go through the list of tasks created and see to it that the user is reminded of those tasks.

2. *Is the requirement really necessary or does it represent an add-on feature that may not be essential to the objective of the system?*

While answering to this question, we thought that the requirement of having a feature to show the billing details was not necessary and hence we eliminated it. Our main objective with this application is to reduce stress of the user by keeping track and reminding of tasks, and providing statistics to maintain efficiency. This feature is general to all people and therefore we did not plan to implement features that would benefit a single set of stakeholders more than another set. Features like automated billing did not apply to a general group of people and hence we removed it.

3. Is each requirement achievable in the technical environment that will house the system or product?

The team decided to remove the feature of 'Group chat' during this phase of requirement collection as it is difficult to achieve in the technical environment that will house the application. The feature would definitely be useful in our application however due to our strict deadline and lack of required skills we discarded this feature.

4. Do any requirements conflict with other requirements?

We surely did come across a few ambiguous and a few out of scope requirements, but our team found out that there were no requirements that conflicted the other. Validation of requirements through this question was done easily.

5. Does each requirement have attribution? That is, is a source (generally, a specific individual) noted for each requirement?

During the process of requirement collection, our team carefully noticed and made a note of the source of each requirement. This has been successfully reflected in the elicitation phase of the requirement collection. Each client has been attributed to a specific set of requirements.

5. Is the specification structured in a way that leads to easy understanding, easy reference, and easy translation into more technical work products?

A Software Requirements Specification(SRS) document has been created to allow easy understanding of the models and functions of the project, along with formal diagrams, descriptions and other graphical models. The SRS is a part of this report and has been included in previous sections.

These were the questions asked and answered to ensure that the requirements model was an accurate reflection of stakeholder needs and that it provided a solid foundation for design.

7.1 Management and Trace Matrices

Requirement Management can be defined as a process of eliciting, documenting, organizing, and controlling changes to the requirements.

We started this process of management of requirements as soon as the elicitation phase began. Our team had a closer look at the requirements collected and had a clear understanding of what the requirements were, and also got an abstract idea of how these requirements can be implemented into actual software, which is just a mobile application in this case.

During this phase, we had a better control on our project. We followed a particular pattern for managing the requirements. What we did first was, identified each of the requirements clearly. This was possible because our team maintained a note of all requirements collected. Then we went through the list and reassembled the requirements based on a priority given by the stakeholder. During this process of reassembling, we also made sure that the requirements were reassembled taking into consideration, the various problems we might face during the implementation of the project.

After having maintained the list of all requirements, we performed Requirement Tracing. The main objective of Requirement tracing was to ensure that all requirements were well understood. [2]

Requirement Tracing is a medium to trace requirements from the start of development process till the software is delivered to the user. This traceability information was stored in a matrix called Tracing Matrix. In the matrix, each requirement was entered in a row and column of the matrix. The dependencies between different requirements were represented in the intersection of the row and column. The following notations were used:

U - Indicates the dependencies of the requirements in the row on the one in the column

R - Indicates the existence of some weaker relationship between the requirements.

For better understanding, an example matrix is given below.

Req ID	1	2	3	4	5
1	-	-	-	-	-
2	-	-	-	U	-
3	-	R	-	U	R
4	U	-	-	-	-
5	-	R	-	-	-

Table 1: This is an example trace matrix

The above example matrix shows that the requirements with unique identifiers 2, 3, and 4 are dependent on the requirements with unique identifiers 3, 4 and 1 respectively. This matrix also shows that there exists some weaker relation between the requirements with

unique identifier 3 and those with identifiers 2 and 5. Similarly the requirement with id 5 has some weaker relation with requirement having an id of 2.

7.2 Requirements List

The requirements were uniquely identified in the list as mentioned above by assigning them with unique identifiers. The following are the identifiers assigned to various requirements collected:

1. Creation of tasks list and monitoring the same
2. Assigning priorities to the tasks
3. Ordering the tasks
4. Setting custom reminders for notifications throughout the day.
5. Creation of group task lists with an administrator and multiple group members
6. Moderating the group list from anywhere, administrator can assign tasks to each member
7. Creation of a routine tasks list that can be kept constant.
8. Statistics for overall task progress for the current day
9. Reminder to set the tasks for the next day.

The traceability matrix designed by our team is shown below:

Req ID	1	2	3	4	5	6	7	8	9
1	-	-	-	-	-	-	-	-	-
2	U	-	-	-	-	-	-	-	-
3	-	U	-	-	-	-	-	-	-
4	R	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-
6	-	-	-	-	U	-	-	-	-
7	U	-	-	-	-	-	-	-	-
8	U	-	-	-	-	-	-	-	-
9	-	-	-	R	-	-	-	-	-

Table 2: An illustration of current requirements trace matrix

The following photos were taken during the requirement gathering process

8.1 Entrepreneurs



Figure 1: Meeting with Rhythm N Brews owner



Figure 2: Meeting with Rhythm N Brews owner



Figure 3: Interview with students



Figure 4: Interview with travel enthusiasts



Figure 5: Group meeting discussions



Figure 6: Finalization of agenda

- [1] Google. Google play services and api guidelines. <https://developers.google.com/android/guides/overview>.
- [2] Roger S. Pressman. *Software Engineering: A Practitioner's Approach*. McGraw Hill, 7th edition, 1982.
- [3] Karl E. Wiegers. Ieee srs template. https://web.cs.dal.ca/~hawkey/3130/srs_template-ieee.doc.