**Title of the project**



**Project Team**

|  |  |  |
| --- | --- | --- |
| **Sl. No.** | **Reg. No.** | **Student Name** |
| **1** |  |  |
| **2** |  |  |
| **3** |  |  |
| **4** |  |  |

**Supervisors:**

**August – 2018**

**B. Tech. in Computer Science and Engineering**

**Faculty of ENGINEERING AND TECHNOLOGY**

**M. S. Ramaiah University of applied sciences**

**Bengaluru -560 054**

**Faculty of engineering and technology**



**Certificate**

*This is to certify that the Project titled “Development of Smart To- Do-List Application” is a bonafide work carried out in the Department of Computer Science and Engineering by <team member names……..> bearing Reg. No. ……………, ………, ……, …………. respectively in partial fulfilment of requirements for the award of B. Tech. Degree in Computer Science and Engineering of Ramaiah University of Applied Sciences.*

**August – 2018**

**<Supervisor name>**

**<HoD CSE Name> <Dean FET Name>**

**Professor and Head – Dept. of CSE Professor and Dean-FET**

Declaration

*Development of Smart To- Do-List Application*

The project work is submitted in partial fulfilment of academic requirements for the award of B. Tech. Degree in the Department of Computer Science and Engineering of the Faculty of Engineering and Technology of Ramaiah University of Applied Sciences. The project report submitted herewith is a result of our own work and in conformance to the guidelines on plagiarism as laid out in the University Student Handbook. All sections of the text and results which have been obtained from other sources are fully referenced. We understand that cheating and plagiarism constitute a breach of University regulations, hence this project report has been passed through plagiarism check and the report has been submitted to the supervisor.

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl. No.** | **Reg. No.** | **Student Name** | **Signature** |
| **1** |  |  |  |
| **2** |  |  |  |
| **3** |  |  |  |
| **4** |  |  |  |

**Date : XX August 2018**

# Acknowledgements

It is with extreme pleasure and pride that we present our B-Tech. dissertation titled “***………………..***”. We would like to express our sincere thanks and gratitude to the following people, who stood by us throughout, helping us with much required inputs, guidance, knowledge and supported us.

We take great pleasure to express our sincere thanks and gratitude to academic project guide <……………>**.** Asst. Professor Department of CSE, for his support, guidance and suggestions throughout the project which is leading this project for the completion.

We express our sincere thanks to**, Dr. M. Arulanantham**, our respected Dean and to **Dr. Raghavendra Venkatesh Kulkarni,** Head of Department of Computer Science and Engineering, for their kind cooperation and support toward out dissertation, and to the management of Ramaiah University of Applied Science for their continued support. We are thankful to the staff members of the Computer Science and Engineering, RUAS for giving us good support and suggestion.

Lastly, we would like to thanks our parents and friends for their continued support, encouragement and motivation and God for paving our way of success in this object.

# Abstract

generated based on the to-do list items. By default, a smart button is available against smart to-do items for further assistance within the application. This project also add functionalities like manual or automatic checking and unchecking of to-do items depending on whether the task is smart or not, providing manual or automatic reminder facility depending on extracted information for the text present in the item etc. leading to a virtual assistance.

The scope of this project is that it is limited to IOS and android platform. Also there are only selected categories for which this application will behave smartly. However, the

**Table of Contents**

[Acknowledgements ii](#_Toc515793673)

[Abstract iii](#_Toc515793674)

[List of Figures vii](#_Toc515793675)

[List of Tables ix](#_Toc515793676)

[1. Introduction 1](#_Toc515793677)

[1.1 Introduction 1](#_Toc515793678)

[1.2 Literature Survey 2](#_Toc515793679)

[1.3 Conclusion 3](#_Toc515793680)

[2. Background Theory 4](#_Toc515793681)

[2.1 Background Theory: 4](#_Toc515793682)

[2.1.1 Java 4](#_Toc515793683)

[2.1.2 Android 4](#_Toc515793684)

[2.1.3 Dart 5](#_Toc515793685)

[2.1.4 Flutter 5](#_Toc515793686)

[2.1.5 Android studio 6](#_Toc515793687)

[2.1.6 Wit 6](#_Toc515793688)

[2.1.7 Natural Language Processing (NLP) 7](#_Toc515793689)

[2.2 Background of Existing Application 7](#_Toc515793690)

[2.3 Conclusion 9](#_Toc515793691)

[3. Aim and Objectives 10](#_Toc515793692)

[3.1 Title 10](#_Toc515793693)

[3.2 Aim 10](#_Toc515793694)

[3.3 Objectives 10](#_Toc515793695)

[3.4 Functional Requirements 11](#_Toc515793696)

[3.5 Method and Methodology 11](#_Toc515793697)

[3.6 Conclusion 13](#_Toc515793698)

[4. Problem Solving 14](#_Toc515793699)

[4.1 Design 14](#_Toc515793700)

[4.1.1 Use Case Diagram 14](#_Toc515793701)

[4.1.2 Sequence Diagrams 16](#_Toc515793702)

[4.2 Implementation 26](#_Toc515793703)

[4.3 Testing 42](#_Toc515793704)

[4.4 Performance Analysis 43](#_Toc515793705)

[4.1.1 Battery consumption 43](#_Toc515793706)

[4.1.2 Response time 57](#_Toc515793707)

[5. Results 61](#_Toc515793708)

[5.1 Screenshot 61](#_Toc515793709)

[5.1.1 Blank To- Do List 61](#_Toc515793710)

[5.1.2 Dark Theme 62](#_Toc515793711)

[5.1.3 Editor Window 63](#_Toc515793712)

[5.1.4 To-do list with existing To-do 64](#_Toc515793713)

[5.1.5 Notification and Bubble 65](#_Toc515793714)

[5.1.6 Contact Details and Number selection 66](#_Toc515793715)

[5.1.7 Call Interface 67](#_Toc515793716)

[5.1.8 Automatic Marking of Check Box 68](#_Toc515793717)

[5.2 Summary 68](#_Toc515793718)

[6. Project Costing 69](#_Toc515793719)

[6.1 Project Cost Estimation 69](#_Toc515793720)

[6.2 Conclusion 69](#_Toc515793721)

[7. Conclusions and Suggestions for Future Work 70](#_Toc515793722)

[7.1 Conclusion 70](#_Toc515793723)

[7.2 Suggestion for future work 71](#_Toc515793724)

[References 72](#_Toc515793725)

[Appendix 73](#_Toc515793726)

# List of Figures

[Figure 1 Use case Diagram 15](#_Toc515793635)

[Figure 2 Low Level Sequence Diagram for call using smart tasker. 16](#_Toc515793636)

[Figure 3 Low Level Sequence Diagram for message using smart tasker 17](#_Toc515793637)

[Figure 4 Low Level Sequence Diagram for Email Using Smart Tasker 18](#_Toc515793638)

[Figure 5 Low Level Sequence Diagram for Call Using Popup 19](#_Toc515793639)

[Figure 6 Low Level Sequence Diagram for message using popup 20](#_Toc515793640)

[Figure 7 Low Level Sequence Diagram for email using popup 21](#_Toc515793641)

[Figure 8 Low Level Sequence Diagram for deleting a todo 22](#_Toc515793642)

[Figure 9 Low Level Sequence Diagram for deleting all todo 23](#_Toc515793643)

[Figure 10 Low Level Sequence Diagram for deleting done todo 24](#_Toc515793644)

[Figure 11 Low Level Sequence Diagram for changing the theme 25](#_Toc515793645)

[Figure 12 Evernote application with set reminder 44](#_Toc515793646)

[Figure 13 Status of Evernote Application 45](#_Toc515793647)

[Figure 14 Google Keep application with set reminder 46](#_Toc515793648)

[Figure 15 Status of Google Keep Application 47](#_Toc515793649)

[Figure 16 Microsoft todo application with set reminder 48](#_Toc515793650)

[Figure 17 Status of Microsoft todo Application 49](#_Toc515793651)

[Figure 18 Todoist application with set reminder 50](#_Toc515793652)

[Figure 19 Status of todoist Application 51](#_Toc515793653)

[Figure 20 Wunderlist application with set reminder 52](#_Toc515793654)

[Figure 21 Status of Wunderlist Application 53](#_Toc515793655)

[Figure 22 todo smartly application with set reminder 54](#_Toc515793656)

[Figure 23 Status of todo smartly Application 55](#_Toc515793657)

[Figure 24 displaying notification bar with reminders of different application 56](#_Toc515793658)

[Figure 25 showing analysis of Switching from light to dark mode 58](#_Toc515793659)

[Figure 26 showing analysis of Switching from dark to light mode 58](#_Toc515793660)

[Figure 27 showing analysis of FPS foreground 59](#_Toc515793661)

[Figure 28 showing analysis of FPS background 59](#_Toc515793662)

[Figure 29 showing analysis of timing graph for notification popup 60](#_Toc515793663)

[Figure 30 showing analysis of constant tapping on theme button 60](#_Toc515793664)

[Figure 31 Blank front view of the application 61](#_Toc515793665)

[Figure 32 showing dark mode of application 62](#_Toc515793666)

[Figure 33 showing editors window 63](#_Toc515793667)

[Figure 34 showing todo with existing todo 64](#_Toc515793668)

[Figure 36 showing notification through notification bar and bubble 65](#_Toc515793669)

[Figure 37 showing number selection interface 66](#_Toc515793670)

[Figure 38 showing calling interface 67](#_Toc515793671)

[Figure 39 showing marking of text 68](#_Toc515793672)

# List of Tables

[Table 1 Literature Survey 2](#_Toc515793617)

[Table 2 Methods and Methodology 11](#_Toc515793618)

[Table 3 Test Cases 42](#_Toc515793619)

[Table 4 Summarised table for battery consumption 57](#_Toc515793620)

[Table 5 Cost estimation table 69](#_Toc515793621)

[Table 6 Specification table for FR 1 73](#_Toc515793622)

[Table 7 Specification table for FR 2 73](#_Toc515793623)

[Table 8 Specification table for FR 3 73](#_Toc515793624)

[Table 9 Specification table for FR 4 74](#_Toc515793625)

[Table 10 Specification table for FR 5 74](#_Toc515793626)

[Table 11 Specification table for FR 6 74](#_Toc515793627)

[Table 12 Specification table for FR 7 75](#_Toc515793628)

[Table 13 Specification table for FR 8 75](#_Toc515793629)

[Table 14 Specification table for FR 9 75](#_Toc515793630)

[Table 15 Specification table for FR 10 76](#_Toc515793631)

[Table 16 Specification table for FR 11 76](#_Toc515793632)

[Table 17 Specification table for FR 12 76](#_Toc515793633)

[Table 18 Specification table for FR 13 77](#_Toc515793634)

# 1. Introduction

In this chapter, the project is introduced with its theme followed by its purpose. Then a literature survey is documented stating most popular existing applications under domain of current project and their key features. In the end, current project and its output will be compared with the existing application and in the end, key features of current project will be evaluated.

## 1.1 Introduction

In a world where everything has become globally connected, information in the form of data is the one which passes in this connection most often. Since humans are the one present on either side of this connection, it won’t be wrong to say that every individual is hit by tremendous amount of information every day. Some information among all can simply be ignored but some information are crucial and can be of at most importance. These information can sometimes lead to tasks that are needed to be accomplished. Now, remembering all tasks and performing them subsequently can become a hectic job especially when the number of tasks are more. For this, people earlier used to write down tasks that they were required to execute and complete accordingly.

Well, living in the 20th century helps, because now due to present technology, this can be done via applications installed on variety of gadgets. Application build for this specific domain are generally knows as to- do list applications. And when it comes to application, a whole new world of innovations explodes. Hence, instead of just listing tasks, other functionality like reminder, alarm, assistance etc. can also be provided via these apps ensuring completion.

In this project the aim is to take the same concept to next level via introducing artificial intelligence into the application and take decisions on various tasks in order to provide extensive assistance.

## 1.2 Literature Survey

Below is the tabulated study of Papers available on this domain. The results are as follows.

Table 1 Literature Survey

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **S**  **l**  **N**  **o.** | **Author(s)** | **Journal Name and year of publication** | **Research Focus** | **Findings in Research** | **Conclusions derived via authors** | **Limitations in the Study** | **Conclusion of Published Work** |
| 1 | [Scott A. Jenson](https://patents.google.com/?inventor=Scott+A.+Jenson) | Schedule and to-do list for a pen-based computer system, 1996 | controlling a schedule and a to-do list on a computer display | This can be done with C++ using open GL library too. | This application can be made via providing accessibility permission in OS. | The technology is slow and not cost effective. | New powerful scheduler was made compared to others. |
| 2 | [Mark R. Hufnagel](https://patents.google.com/?inventor=Mark+R.+Hufnagel)  [Tyler S. Lacock](https://patents.google.com/?inventor=Tyler+S.+Lacock)  [Aaron K. Shook](https://patents.google.com/?inventor=Aaron+K.+Shook) | Auto-generated to-do list | Methods, systems, and computer readable media for providing an auto-generated to-do list | New system level technology as WED to make it better | It was one of the best auto generated to-do list. | It was not marketed well because of its non-user friendly nature. | Text is analysed to identify the task sender, the task body, and the task date. |
| 3 | [Hikaru Suzuki](https://patents.google.com/?inventor=Hikaru+Suzuki) | Interactive to-do list item notification system including GPS  interface | Used new GPS systems available in the market for investigation. | Placing infrared and magnetic devices for better list. | The application can be made to reach new heights via use of sensors in devices.. | Cannot be used in conditions when another activity automatically starts. Example calling. | It was the first to indulge in use of GPS in to – do list application. |

## 1.3 Conclusion

In this chapter, different to do list available in the market were discussed extensively with their features so that the proposed application can have a good comparison to what is available in the market. Also technical papers with respect to to-do list application were surveyed in order to get the knowledge of past work done in this domain. This will help in getting a clear idea of innovation associated with this proposed project.

# 2. Background Theory

In this chapter, all theory related to proposed project including technical aspect and resources are explained. This section starts with discussion of Programming Languages and Framework Used, IDE’s on which the proposed project is built. Then discussion will turn towards online Artificial Intelligent Engines available and the one Used in this project. In the end the specific domain in which these engines are used will be explained. In the end, the entire theory will be summarised via a summery.

## 2.1 Background Theory:

This section gives complete knowledge and understanding of different technologies and framework that were required in this project for its completion. The background Theory is listed below:

### 2.1.1 Java

Java is a programming language that is based on object oriented modelling of the world. This means that all entities are treated as objects and are maintained in a hierarchal manner in order to correlate each object and reduce redundancy in properties. It has many concepts like inheritance, encapsulation, polymorphism, abstraction etc. based on which blueprints of objects, i.e. classes are structured. Knowledge of Java plays an important role since it’s a pure object oriented programming language thus making it optimum to learn object oriented paradigm. This also helps in learning other languages that are based on equivalent paradigm.

### 2.1.2 Android

Android is a software and an operating system based on Linux with its Own SDK (Android SDK). Android is a product of Google followed by Open Handset Alliance (California). Features of android includes an open source platform which means that anyone can customise it based on his/her requirement. It provides interface to interact with hardware

present in the embedded system for a more complex and synchronised use in order to achieve expanding and sophisticated needs.

Since, an application was the base of this project, hence android was necessary to build the application from scratch satisfying all expected objectives.

### 2.1.3 Dart

Dart is a general purpose programming/coding language which is developed by google and approved by ECMA. The main purpose of this language is to build web, mobile and server applications. It is also used for programming in IOT (Internet of Things) domain. It is open source with free permission software licence.

Dart is based on object oriented programming concept, with defined classes supporting mostly single inheritance.

This is necessary since Flutter, the UI framework supported by google is based and is manipulated by this language. Since, flutter, used this language, learning Dart language became necessity in order to complete the proposed project.

### 2.1.4 Flutter

Flutter is a UI framework developed by Google in order to make its UI design simpler using its Material Design Pattern. It has its own SDK which helps in crafting high quality user interface. This technology automatically generates interface for iOS and Android platform using same codebase i.e. Dart in record time. It is still in beta state but has become extremely popular among developers because of its extended ease towards implementation of Material Design along with its fee and open source nature. A great feature of flutter in terms of its execution is Hot Reload, which helps in quickly reflecting changes to UI with change in code instead of reinstalling the application every time(which is the case in normal android development). In Flutter, Layered architecture of UI allows developer to fully customize, which results in incredibly fast rendering and expressive and flexible designs. As already mentioned, since flutter supports UI of both iOS and Android Platform, critical platform differences in scrolling, navigations, icons, fonts etc. are automatically taken care of. This was the reason why flutter was chosen for this project in order to develop the application required for the project in both the platforms.

### 2.1.5 Android studio

Android Studio is an IDE i.e. an Integrated Development Environment designed specifically for development of android application. This was developed in order to provide a utility rich environment for android developers for both development and testing of applications. This IDE comes with built in support for Google Cloud Platform making it easier for integration of google Cloud Messaging and Application Engine, AVD manager which provides feature rich emulator, extensive frameworks and testing tools etc. This IDE also comes with flexible Grade- Based build system with is important for build an Android Application.

This IDE was used for development of application proposed in this project. This IDE helped in debugging the code faster and providing suggestion for quick and error free coding.

### 2.1.6 Wit

Wit is a platform provided by Facebook that help developers to create a natural Language Understanding Bot. it used concept of deep learning and used its engine to understanding natural Language text and give reply accordingly. Wit is trained using examples and with increasing in number of example, the ability of wit to understand text under developers expected domain also increases. In an example, entities are needed to be marked of wit to understand the sentence. There are several entities like, contact, phone number, message etc. that are trained on millions of examples but it also provides facility to create personal entities and train them via examples. Wit can also detect the emotion of the statement via its deep learning algorithm. After the engine being trained, the reply can be traced in the form of JSON output which could further be used via different applications to satisfy their purpose.

Wit was required in this project in order to understand the input given by the user and then extract key information from the sentence in order to satisfy the requirement of the project.

### 2.1.7 Natural Language Processing (NLP)

NLP is a part of computer science dealing with the interaction between computer and human’s natural language. In other words, it is the study of ability of computers to understand natural way of language delivery efficiently. The different challenges involved in this field are speech recognition, natural language generation and understanding. This is done via sampling the received sound wave and matching frequency of each sample with existing sample frequency. There is always a tolerance level because of difference of pitch and texture of voice, noise etc.

For this project, Natural language understanding was more important since information was needed to be extracted from sentences directly given by the user. This was accomplished via using wit engine developed by Facebook.

## 2.2 Background of Existing Application

For an individual, the best to-do application is the one which works for him. In present market, there are many to-do list applications that can be downloaded from application store or are pre built into a person’s phone. Most of the applications (including applications in this particular domain) are free whereas few of them are paid. Different to-do lists have different ways of execution along with overlapping features. This is the reason why sometimes it becomes difficult to select the best to-do list application among many in the present market. Hence, few of the popular applications are mentioned below with their key features listed briefly. This is done in order to know the existing application in this domain along with their features so that the proposed project can have adequate references and the work done in this project can imply something new and innovative which can further contribute in the advancement of this domain.

* **Google Keep**
  1. Google keep is faster and lighter than most of the application present today in current market.
  2. Voice note feature is available for instant notes which is also helpful for handicap people.
  3. It also features google drive sync hence making data portable which can be accessed via any platform.
  4. Google keep’s webpage is fast and has a minimalistic design to keep the overall experience simple.
  5. Google keep also has its own widget which has been appreciated via critics because of its fabulous design.
  6. It also features collaboration list that can help multiple people maintain single list for shared notes.
* **Todoist**
  1. It uses natural language processing in order to take lightning fast input from audio source.
  2. It allows labels and filters for faster search of notes and to- do list.
  3. Collaboration feature is also available enabling people to come together on a single list.
  4. It has an attractive minimalistic design.
  5. This application is available across all platforms like Android, IOS etc.
  6. There are 3 different table in which todoist organises its todo: **Inbox**, for stuff a person has not completed yet; **Today**, for things to be done today; and **next 7 Days**, for the coming week.
  7. It allows syncing of tasks into google calendar for better view of tasks.
* **Wonderlist**

1. Got app of the year award by apple since 2013
2. Collaboration feature is available enabling people friends, family, or co-workers to come together on a single list.
3. The design of it is sleekly designed along with being user-friendly.
4. Adding, checking and unchecking of list items is simple and intuitive.

* **Evernote**
  1. It has a plugin for browsers that helps in directly clipping stuff from internet and saving it on the list.
  2. Evernote provides a private email address to every user which helps in sharing any note via email super easy.
  3. This application provides merging feature for users that make multiple notes and late want to merge them together.
  4. This application keeps security into concern via providing a two-step verification feature in order to keep a person’s research safe.
  5. One desktop version of Evernote, the user can set up encryption for notes as well as sections of notes i.e. text for further security.
* **Microsoft to- do**

1. It has the fastest sync compared to all other to do list application available in market since 2018.
2. There is no collaboration feature in Microsoft to- do but it has Streamlined task feature.
3. This application is free without optional upgrade via payment.

## 2.3 Conclusion

In this Chapter, All background knowledge for completion of this project are explained elaborately. Also, in each mentioned resource and technology, the reason for its use is also expressed. Languages like Java, Dart and its use in this project is explained. The reason for Using Flutter has been covered extensively. In the end, advantages of using Android studio IDE is explained along with the use of NLP in wit’s AI Engine is covered completely.

# 3. Aim and Objectives

This chapter focuses on defined title and Aim of the project correctly and clearly. Later this chapter also includes the required objectives that needed to be fulfilled in order to complete this project. Functional Requirements are well documented in this section since it is required to design different diagrams leading to complete view of the project. This is followed by method and methodologies that tabulates the procedure at will be followed in order to complete the objectives. This section then ends with a summary.

## 3.1 Title

Design and Development of a Smart-To-Do list Application.

## 3.2 Aim

To design and develop a smart to-do-list application for Android devices using Java technologies and current popular Framework.

## 3.3 Objectives

The objectives of the proposed Project are listed below:

1. To conduct a literature survey on the existing to-do-list applications, related methods and techniques.
2. To arrive at requirements for the smart to-do-list application based on the literature survey.
3. To arrive at the design specifications of the application based on the identified requirements.
4. To implement the design using Object oriented approach.
5. To test the developed application and study its performance.
6. To write the project report.

## 3.4 Functional Requirements

The functional Requirements for this project are mentioned below:

1. The user must be able to add a to-do item.
2. The user must be able to delete a to-do item.
3. The user must be able to edit a to-do item.
4. The user must be able to set reminder for a to-do item.
5. The user must be able to delete all to-do item.
6. The user must be able to delete only done to-do item.
7. The user must be able to change the theme of the application.
8. The user must be able to call a person in to-do item using smart button.
9. The user must be able to message a person in to-do item using smart button.
10. The user must be able to email a person in to-do item using smart button.
11. The user must be able to call a person in to-do item via automatic popups.
12. The user must be able to message a person in to-do item via automatic popup.
13. The user must be able to email a person in to-do item via automatic popup.

## 3.5 Method and Methodology

Table 2 Methods and Methodology

|  |  |  |  |
| --- | --- | --- | --- |
| **Objective No.** | **Statement of the Objective** | **Method/ Methodology** | **Resources Utilised** |
| 1 | To conduct a literature survey on the existing to-do-list applications, related methods and techniques | 1.1 Literature review on to-do-list system will be carried out by referring reviewed journals, books, manuals, related documents, by interviewing users of the existing applications.  1.2 Based on the survey, the characteristics of a to-do-list system and related methodologies will be documented. | IEEE project paper and reputed websites along with Books from orally Publications. |
| 2 | To arrive at requirements for the smart to-do-list application based on the literature survey. | 2.1 Based on the characteristics identified in the survey, a generic list of requirements for a smart to-do-list system will be generated.  2.2 The functional and non-functional requirements of the smart to-do-list application will be derived after conducting a risk analysis on the feasibility of development effort with respect to time, manpower, cost and technology.  2.3 An analysis model of the application will be developed using UML use case and sequence diagrams.  2.4 A Software Requirement Specification (SRS) document will be created. | Stack overflow Website along with the help of mentor. |
| 3 | To arrive at the design specifications of the application based on the identified requirements. | 3.1 Based on the SRS document, a high level design specification will be created using UML sequence. | Get help of mentor and drawing diagrams using DIA. |
| 4 | To implement the design using Object oriented approach. | 4.1 Based on the design specification, appropriate off-the-shelf libraries such as to-do txt-touch will be selected to reduce development time.  4.2 The design of the to-do-list application will be implemented using Java and Flutter programming language on the Android Studio IDE. | Android studio platform along with Flutter UI Framework and their official website. |
| 5 | To test the developed application and study its performance. | 5.1 The software will be tested for passing of all functional requirements.  5.2 Performance analysis of the application will be conducted to ensure efficient usage of battery and acceptable response time. | Android studio platform and android developer website. |
| 6 | To write the project report. | 6.1 Develop a scientific project report as per the template specified.  6.2 Display and demonstrate in University organized project exhibition. | Standard Template of Document provided by University. |

## 3.6 Conclusion

This particular chapter focused on defined title and Aim of the project correctly. Later this chapter included required objectives that were required to be fulfilled to complete this project. Functional Requirements is documented successfully in this section followed by method and methodologies which is successfully tabulated in order to know the steps used in completing the objectives including resources used.

# 4. Problem Solving

In this section, the actual dissection of project is done and each module is built piece by piece in order to complete the project. In design section, the application has been built in accordance with functional requirements based on which diagrams like Use Case and Low Level Sequence Diagram is drawn. In implementation section, snips of important code is displayed with their explanation given below. In testing section, all functional requirements are tested and the result is analysed resulting in status of test condition. In the ending section, performance analysis is done on various factors such as battery, booting time and time requirement for major operation.

## 4.1 Design

Design is necessary when it comes to development since it acts as a blueprint for entire process from requirement making to finished (final product). Hence, in this section designs specific to this project like use case, sequence diagram etc. are attached.

### 4.1.1 Use Case Diagram

….

## 4.2 Implementation

Explain Here….

## 4.3 Testing

All functional requirement are tested and are tabulate below:

Table 3 Test Cases

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl No.** | **FR No.** | **Expected** | **Obtained** | **Result** |
| 1 | FR 1 | The user must be able to add a to-do item. | The user was successfully able to add a to-do item. | PASS |
| 2 | FR 2 | The user must be able to delete a to-do item. | The user was successfully able to delete a to-do item. | PASS |
| 3 | FR 3 | The user must be able to edit a to-do item. | The user was successfully able to edit a to-do item. | PASS |
| 4 | FR 4 | The user must be able to set reminder for a to-do item. | The user was successfully able to set reminder for a to-do item. | PASS |
| 5 | FR 5 | The user must be able to delete all to-do item. | The user was successfully able to delete all to-do item. | PASS |
| 6 | FR 6 | The user must be able to delete only done to-do item. | The user was successfully able to delete only done to-do item. | PASS |
| 7 | FR 7 | The user must be able to change the theme of the application. | The user was successfully able to change the theme of the application. | PASS |
| 8 | FR 8 | The user must be able to call a person in to-do item using smart button. | The user was successfully able to call a person in to-do item using smart button. | PASS |
| 9 | FR 9 | The user must be able to message a person in to-do item using smart button. | The user was successfully able to message a person in to-do item using smart button. | PASS |
| 10 | FR 10 | The user must be able to email a person in to-do item using smart button. | The user was successfully able to email a person in to-do item using smart button. | PASS |
| 11 | FR 11 | The user must be able to call a person in to-do item via automatic popups. | The user was successfully able to call a person in to-do item via automatic popups. | PASS |
| 12 | FR 12 | The user must be able to message a person in to-do item via automatic popup. | The user was successfully able to message a person in to-do item via automatic popup. | PASS |
| 13 | FR 13 | The user must be able to email a person in to-do item via automatic popup. | The user was successfully able to email a person in to-do item via automatic popup. | PASS |

## 4.4 Performance Analysis

# 5. Results

# 6. Project Costing

This chapter deals with costing of this project which gives an overall estimation of expenses that was required to complete this project. This Covers expenses of testing devices, Platform and Hardware cost, Human Resource Cost and a grand total of Entire cost.

## 6.1 Project Cost Estimation

The cost of project is summarised in a tabular form displayed below:

Table 5 Cost estimation table

|  |  |  |
| --- | --- | --- |
| **Serial Number** | **Resources and Work** | **Cost(Rs)** |
| 1 | Three Smartphones with each cost 5K (for testing) | 15,000/- |
| 2 | Laptop (for development) | 30,000/- |
| 3 | Human Resources (4 \* 15,000/-) | 60,000/- |
|  | **TOTAL** | **1,05,000/-** |

## 6.2 Summary

Since this project didn’t have any physical model, hence no expenses were made for physical model. However, effort on making the software via parallel learning of new technology raised the Human Resource cost which can be seen in Table 5.

Three Smartphones with different configuration and platform were chosen for testing the application. The test was completed successfully.

# 7. Conclusions and Suggestions for Future Work

In this chapter, conclusion has been given for entire project along with conclusion to each section present in the report. All are explained with status of completion of each section mentioned clearly. This section ends with Suggestion and scope of future work which direct this project towards new openings of technology where the same project can be extended in order to meet the requirement of customers time to time.

## 7.1 Conclusion

This project started with Literature survey done via gathering information from different IEEE papers, patented documents and reputed Websites. Books from Orally publication were also used as a guide. Also, popular application were listed against their features making it clearer to compare and contrast applications with each other along with application proposed in this project. Background Theory of all resources including technology used, Framework selected, IDE’s worked upon and engines applied were extensively elaborated so that these theory can be applied effectively and the reason for their use/ application can be well understood. Later, all objectives were listed done after declaring title and Aim of the project and methods and mythologies in order to complete the bulleted objectives were well tabulated. From objectives, Functional Requirement was extracted and were well segregated for sequential completion of project. Later, diagrams including Use-case, Sequence, Class, Block and Widget Hierarchy Diagram were created giving complete view of the project from different perspective. Implementation of the project was displayed via displaying code written in both java and Dart programming language interacting with each other to create the application. Testing was done for all functionalities and were found to be working successfully. Later in result section, all screenshot of application in different states were taken in order to demonstrate the end product of this project. Each screenshot was explained with its importance as a view for the application. Performance analysis was done in order to give numerical value to performance of the application clearly stating the advancement in application proposed in this project compared to other applications present in the market. Later project cost estimation was done in order to know the financial asset required to rebuild this project. The entire project was concluded with s suitable conclusion and its scope in near future.

## 7.2 Suggestion for future work

Innovation is a never ending process, hence bring new innovation and extension of this project is always possible. This project for now is limited to 3 smart functionalities i.e. call, message and email but this can be extended for other intents too. For example "buy". If a consumer makes a to-do for buying groceries (Say), the application can be extended in such a way that it can detect actual quantities of each ingredient form a complex formation of text. For example, if the user writes “Buy a kg of apple and Banana”, the application must be able to understand that the quantity 1kg is referred to both apple and banana and not just apple. Also this application can be extended to collect the items and quantities from the to-do item and search for it on online stores and popups the result in the form of notification or a layout listing the items, their quantity and their pricing.

Application like booking a flight, hotel, lounge etc. can also be added to this application. For example, if a user types ”Book cheapest flight from Bangalore to Delhi for tomorrow between 2 to 5 pm”, this sentence contains a lot information regarding the departure location, arrival location, require time and date of booking and a cheap flight filter. This information can be used to come out with results of available airlines tickets which can be shown to the user in the form of a notification or popups.

This application instead of a to-do list application, can also go further to a note taking application. Then features like automatic calculation can also be added. For example, say a user maintains a cash flow with his friend with simple positive and negative signs against the price and a small detail of what that money is for, the application can automatically calculate the total and display it as notification provide calculate total for “so and so name” is listed on the to-do list.

# References

1. Bellotti, V., Dalal, B., Good, N., Flynn, P., Bobrow, D.G. and Ducheneaut, N., 2004, April. What a to-do: studies of task management towards the design of a personal task list manager. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 735-742). ACM.
2. Hufnagel, M.R., Lacock, T.S. and Shook, A.K., International Business Machines Corp, 2012. *Auto-generated to-do list*. U.S. Patent 8,108,206.
3. Jenson, S.A., Apple Inc, 1996. *Schedule and to-do list for a pen-based computer system*. U.S. Patent 5,570,109.
4. Parekh, H. and Delvadiya, D., 2016. TO Do List Application.