RAMAIAH
Institute of Technology

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

A Dissertation Report on

Intelligent Daily Scheduler

Submitted by

Akanksh B S	1MS15CS008
Koushik A S	1MS15CS058

Bachelor of Engineering in Computer Science & Engineering

Under the guidance of

Geetha J
Assistant Professor

RAMAIAH INSTITUTE OF TECHNOLOGY
(Autonomous Institute, Affiliated to VTU)
BANGALORE-560054
2017-2018, www.msrit.edu,

Ramaiah Institute of Technology

(Autonomous Institute, Affiliated to VTU)
BANGALORE-560054

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that the project work titled **Intelligent Daily Scheduler is** a bonafide work carried out by **Akanksh B S(1MS15CS008) and Koushik A S(1MS15CS058)** in partial fulfillment for the course of Bachelor of Engineering in Computer Science and Engineering during the year 2018. The Project report has been approved as it satisfies the academic requirements with respect to the project work. To the best of our understanding the work submitted in this report has not been submitted anywhere.

Signature of the Guide	**Signature of the HOD**
Geetha J	Dr. Anita Kanavalli

External Examiners

Name of the Examiners:	Signature
1.	
2.	

DECLARATION

We Students of sixth semester BE, Dept. of Computer Science and Engineering, Ramaiah Institute of Technology, Bangalore, hereby declare that the project entitled "**Intelligent Daily Scheduler",** thesis completed and written by us under the guidance of **Geetha J,** Dept. of CSE, Bangalore.

Place:
Date:
(1MS15CS008 Akanksh B S)
(1MS15CS058 Koushik A S)

ACKNOWLEDGEMENT

First and foremost, my utmost gratitude to Geetha J dept of CSE, MSRIT whose sincerity and encouragement we will never forget. She has been our inspiration as we overcame all the obstacles in the completion of this project work.

Dr. Anita Kanavalli, Head of the Department of Computer Science and Engineering, had kind concern and consideration regarding project work and we would like to thank her for continues support.

We would like to thank our beloved principal Dr. N.V.R Naidu for his support and encouragement.

This work would not have been possible without the guidance and help of several individuals who in one way or another contributed their valuable assistance in preparation and completion of this study.

We would like to express sincere thanks to all the teaching and non-teaching faculty of CSE Department and my dear friends who helped in all the ways while preparing the Report.

Abstract

It is quite evident that everyone have certain objectives to be completed and quite a few expectations to be met. Accomplishments of these targets requires proper planning in addition to an individual's abilities. Smartphones have seen substantial growth over the past decade and there is perceptible evidence of people relying on them. Although there are solutions in the form of todo list applications to mitigate the scheduling problem, they provide little insight regarding an individual's time organisation. In our application focus is laid on learning the leisure time of the user by indirectly monitoring day to day activities through the app. Free time as predicted by the model and the list of tasks to be performed form the input to the scheduling algorithm. The scheduling algorithm then allocates time for each task and the user is notified of the final schedule.

LIST OF FIGURES

LIST OF TABLES

Contents

1. Introduction

1.1 General Introduction

Every individual has a set of short and long term goals and commitments which are to be executed in a certain period of time. The frequency and varying intensity of these tasks have made people rely on to-do lists and other such reminder applications. It is to be noted that these applications help people to just note down their tasks and also possibly set themselves a predefined notification which just acts a sophisticated alarm. The issue of major concern is that even though all the people in the globe find not more than twenty four hours in a day, some people are quite successful in their lives while others fail to do so which surprisingly is not solely dependent on an individual's talent and potential. These shortcomings can be attributed to improper time management by people which translates to either people overestimating or underestimating the time they have. The consequences of the former being people having too many commitments and unable to fulfill most of them while the latter often occurs as a result of procrastination. In either of the cases the potential of the person is significantly wasted. Many people find it difficult to strike a balance between the extreme cases. It is also notable that there is a certain degree of efficiency which people achieve while they are executing their tasks and this figure drops down quite significantly when the individual tries to over schedule the tasks or work beyond his/her potential. This scenario can be well understood in the case of students[1] who while away most of their time during the start of the academic year and try to complete the entirety of the syllabus in the last few days before the examination. This leads to another important factor of scheduling which is load balancing. All the above said factors vary across individuals and thus cannot be generalized. The solution to the problem requires a considerable degree of personalization.

1.2 Problem Statement

The Daily scheduling problem is defined to find a time table for the week that efficiently assigns the various tasks of users taking into consideration constraints like users' daily routine life. The problem also includes finding free time of users and distribute the tasks fairly throughout the week. This depends upon the available spare time of the user which varies across different days of the week. The problem is a classical scheduling problem. The constraints of scheduling problem are:

- The free time of users should be more than the time required to complete tasks.
- Two or more tasks should not be assigned in a single time interval.
- All tasks should be allocated with sufficient time for completion.
- Tasks should be evenly divided so that no day is overscheduled by regular work or scheduled tasks.
- Tasks should be assigned with minimum interruption.

1.3 Objectives of the project

The main objective of the project is to improve individuals output by providing means to schedule their tasks in an efficient way. We do this by developing a user-friendly app which is supported for all versions of android 4.1(Jelly Bean) or above. This app schedules tasks for users in personalized way without minimum overhead on user. This app also employing Load Balancing schemas and other techniques to ensure maximum efficiency in schedule designed for users.

1.4 Project deliverables

The main project deliverable is the final working mobile application that can schedule the tasks for users in a personalized way. The project should develop a Neural Network with a minimum accuracy of 70% and a scheduling algorithm. The table describes different deliverables obtained as a part of the project.

SL .NO	Deliverable
1	Synopsis and Problem Definition
2	Literature Survey
3	Project Plan
4	System Design
5	Sequence Diagram
6	Scheduling Algorithm
7	Neural Network Model
8	Android app (final product)

Table 1.1 Project Deliverables

1.5 Current Scope

The project falls under the domain of scheduling applications. The aim was to exploit the ubiquity of smartphones and develop a user-friendly android application. Currently the application analyses the individuals’ free time based on their day to day activities. The free time available across the week.

1.6 Future Scope

The app can be further extended and improved by integrating it with Google calendrer and other such platforms. Scheduling algorithm can also be improved by also considering tasks deadline before scheduling tasks. The priority of Tasks can also be considered to make sure that higher propriety tasks are scheduled first than lower priority tasks.

2. Project organization

2.1 Software Process models

We have followed Waterfall software process model to finish the project. We planned and did Literature survey and System design before working on implantation. The figure describes the process of waterfall model.

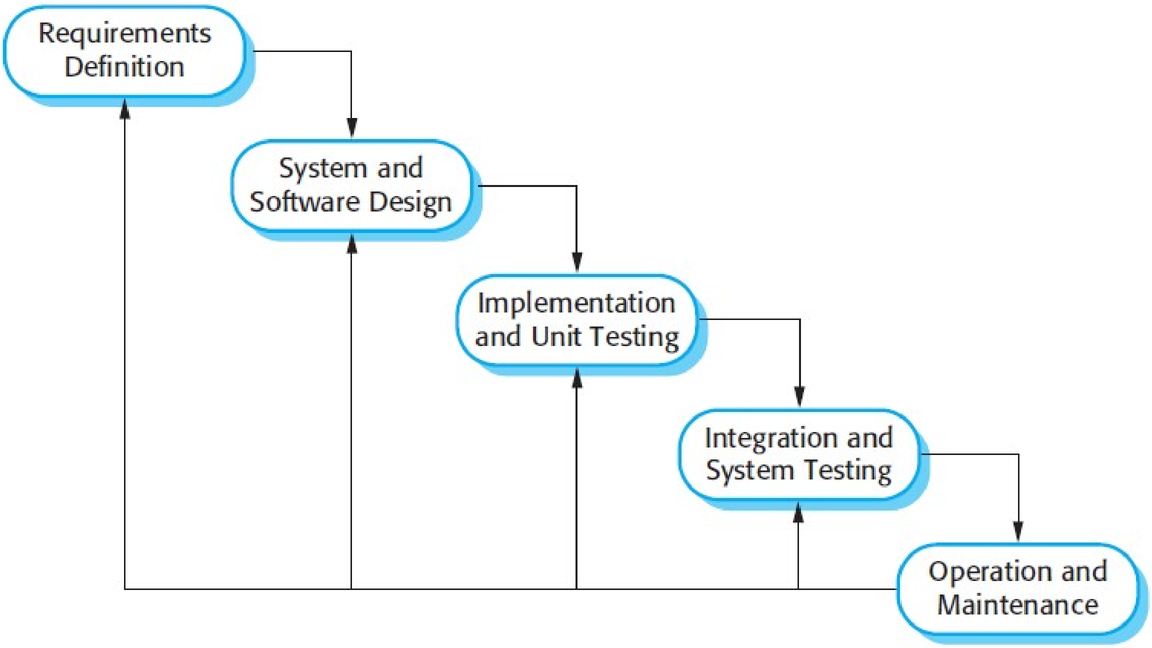

Requirements Definition
System and Software Design
Implementation and Unit Testing
Integration and System Testing
Operation and Maintenance

Fig 1.2 Waterfall Process Model

2.2 Roles and responsibilities

Role / Person	Akanksh B S	Koushik A S
Literature Survey /Requirement Definition	R,A,C,I	R,A,C,I
Neural Network Model	C,I	R,A
Scheduling Algorithm	C,I	R,A
User Interface	R,A	C,I
Testing	R,A	C,I

Table 2.1 Roles and Responsibilities

3. Literature Survey

3.1 Introduction

Good amount of research work has been done in the past decade on Time Management and Scheduling of tasks. However, all this research is limited to only study of time management in various cases and possible solutions. No researcher has developed an application or other methods using these studies to help people better manage their time. We have referred the following work done by the researchers to build a mobile application and schedule tasks using Deep learning methods and Load Balancing Schemas.

3.2 Related Works with the citation of the References

1) Time Management: A Realistic Approach[2] talks about the key steps for successful time management which are 1) set realistic goals 2) get organised 3) delegate 4) relax and recharge and 5) stop feeling guilty. The author stresses the importance of maintaining a to-do list for weekly and life-time goals. Author further goes on to say that key component in staying organized is to maintain a calendar and schedule tasks daily.

2) In Time Allocation and Task Juggling[3] paper the authors have presented a model to study switching cost between tasks and productivity of individuals. They conclude with an observation that working on too many tasks at the same time reduces an individual’s output.

3) Vincent C.S Wiers[4] in his paper studies applicability AI techniques in production scheduling. He believes that humans are a key element in scheduling and success or failure of scheduling depends on humans. Modelling human expertise turned out to be useful to develop more efficient search strategies than would have been possible with operations research techniques. His studies show that there is variability between individuals’ behaviour and it is really hard to schedule for everyone with a single schedule. The question of how to improve this leads us to a discussion about the importance of feedback. Though feedback about the effectiveness of behaviour has long been recognised and appreciated it is essential that such feedback has to be specific and timely to be effective

Vincent C.S Wiers[4] finally concludes with an observation that Integral display facilities turned out to increase performance in scheduling tasks where integration of information is required, i.e., where the amount of task elements to be handled simultaneously is high. In the operations research community, the GANTT chart is generally seen as an effective means to represent scheduling problems to humans. The manual functions and the graphical representation of the production system give the user significant hold over the scheduling process.

4) Recently, [5] Smartphine, a mobile app has been developed which helps students to schedule their calendar according to the student's mood. This is an interesting idea where they try to reduce the stress of students. They track the mood and behavioural analysis of the students through their smartphones. They incentivise and motivate students to study better. However, most of this app is not designed to schedule students or user's tasks in an effective way so that user is not overscheduled or to make them productive.

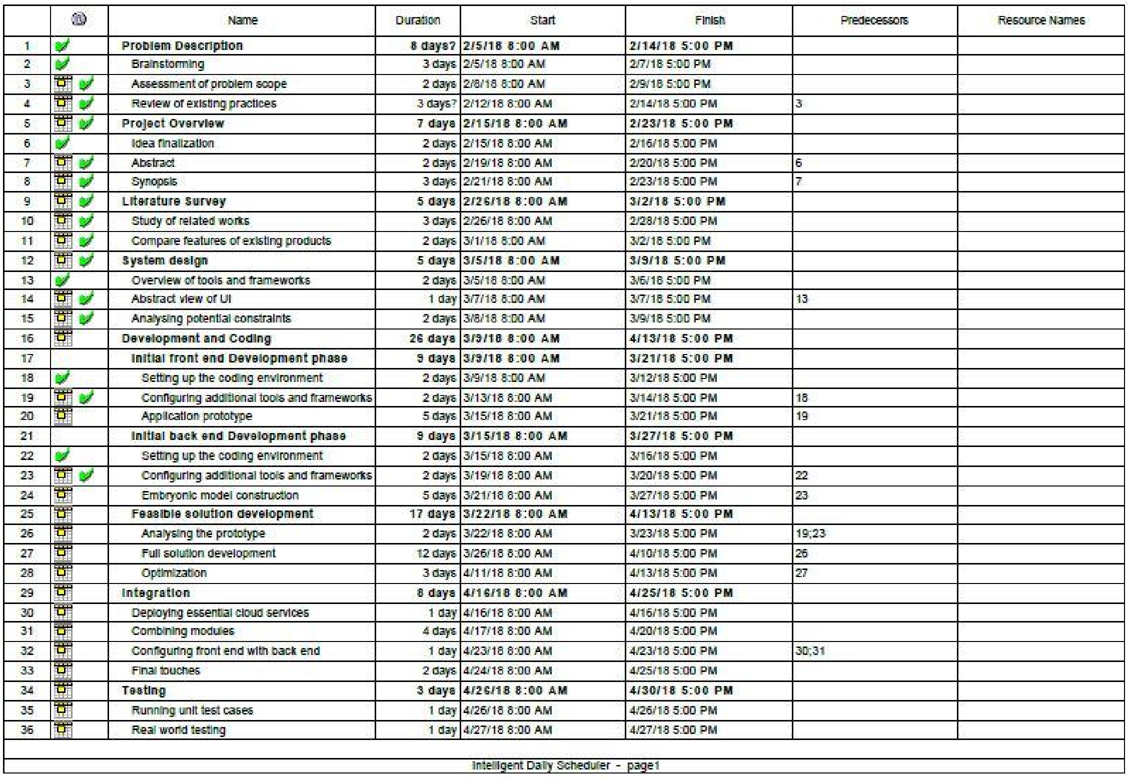
3.2 Conclusion of Survey

From the above Survey we can conclude that having to-do list to maintain a goal and staying organised by scheduling tasks is helpful to increases persons output. Tasks scheduled should be in a such a way that we need to minimize task juggling and allocate continuous long intervals of time rather than small intervals as far as possible. To do this we can apply AI Techniques with timely feedback to schedule tasks in a personalised way. Therefore, we collect feedback continuously and in a timely way through our app so that we can predict and develop the most effective schedule for that user. Finally having a good and easy user interface on their mobiles for users to know their tasks and their daily schedule is crucial for better output of the app.

4. Project Management Plan

4.1 Schedule of the Project

The schedule of the project can be represented in the form of a gantt chart as follows.



	ⓘ	Name	Duration	Start	Finish	Predecessors	Resource Names
1	✓	Problem Description	8 days?	2/5/18 8:00 AM	2/14/18 5:00 PM		
2	✓	Brainstorming	3 days	2/5/18 8:00 AM	2/7/18 5:00 PM		
3	▦ ✓	Assessment of problem scope	2 days	2/8/18 8:00 AM	2/9/18 5:00 PM		
4	▦ ✓	Review of existing practices	3 days?	2/12/18 8:00 AM	2/14/18 5:00 PM	3	
5	▦ ✓	Project Overview	7 days	2/15/18 8:00 AM	2/23/18 5:00 PM		
6	✓	Idea finalization	2 days	2/15/18 8:00 AM	2/16/18 5:00 PM		
7	▦ ✓	Abstract	2 days	2/19/18 8:00 AM	2/20/18 5:00 PM	6	
8	▦ ✓	Synopsis	3 days	2/21/18 8:00 AM	2/23/18 5:00 PM	7	
9	▦ ✓	Literature Survey	5 days	2/26/18 8:00 AM	3/2/18 5:00 PM		
10	▦ ✓	Study of related works	3 days	2/26/18 8:00 AM	2/28/18 5:00 PM		
11	▦ ✓	Compare features of existing products	2 days	3/1/18 8:00 AM	3/2/18 5:00 PM		
12	▦ ✓	System design	5 days	3/5/18 8:00 AM	3/9/18 5:00 PM		
13	✓	Overview of tools and frameworks	2 days	3/5/18 8:00 AM	3/6/18 5:00 PM		
14	▦ ✓	Abstract view of UI	1 day	3/7/18 8:00 AM	3/7/18 5:00 PM	13	
15	▦ ✓	Analysing potential constraints	2 days	3/8/18 8:00 AM	3/9/18 5:00 PM		
16	▦	Development and Coding	26 days	3/9/18 8:00 AM	4/13/18 5:00 PM		
17		Initial front end Development phase	9 days	3/9/18 8:00 AM	3/21/18 5:00 PM		
18	✓	Setting up the coding environment	2 days	3/9/18 8:00 AM	3/12/18 5:00 PM		
19	▦ ✓	Configuring additional tools and frameworks	2 days	3/13/18 8:00 AM	3/14/18 5:00 PM	18	
20	▦	Application prototype	5 days	3/15/18 8:00 AM	3/21/18 5:00 PM	19	
21		Initial back end Development phase	9 days	3/15/18 8:00 AM	3/27/18 5:00 PM		
22	✓	Setting up the coding environment	2 days	3/15/18 8:00 AM	3/16/18 5:00 PM		
23	▦ ✓	Configuring additional tools and frameworks	2 days	3/19/18 8:00 AM	3/20/18 5:00 PM	22	
24	▦	Embryonic model construction	5 days	3/21/18 8:00 AM	3/27/18 5:00 PM	23	
25	▦	Feasible solution development	17 days	3/22/18 8:00 AM	4/13/18 5:00 PM		
26	▦	Analysing the prototype	2 days	3/22/18 8:00 AM	3/23/18 5:00 PM	19;23	
27	▦	Full solution development	12 days	3/26/18 8:00 AM	4/10/18 5:00 PM	26	
28	▦	Optimization	3 days	4/11/18 8:00 AM	4/13/18 5:00 PM	27	
29	▦	Integration	8 days	4/16/18 8:00 AM	4/25/18 5:00 PM		
30	▦	Deploying essential cloud services	1 day	4/16/18 8:00 AM	4/16/18 5:00 PM		
31	▦	Combining modules	4 days	4/17/18 8:00 AM	4/20/18 5:00 PM		
32	▦	Configuring front end with back end	1 day	4/23/18 8:00 AM	4/23/18 5:00 PM	30;31	
33	▦	Final touches	2 days	4/24/18 8:00 AM	4/25/18 5:00 PM		
34	▦	Testing	3 days	4/26/18 8:00 AM	4/30/18 5:00 PM		
35	▦	Running unit test cases	1 day	4/26/18 8:00 AM	4/26/18 5:00 PM		
36	▦	Real world testing	1 day	4/27/18 8:00 AM	4/27/18 5:00 PM		

Intelligent Daily Scheduler - page1

Fig 4.1 Project Plan Spreadsheet

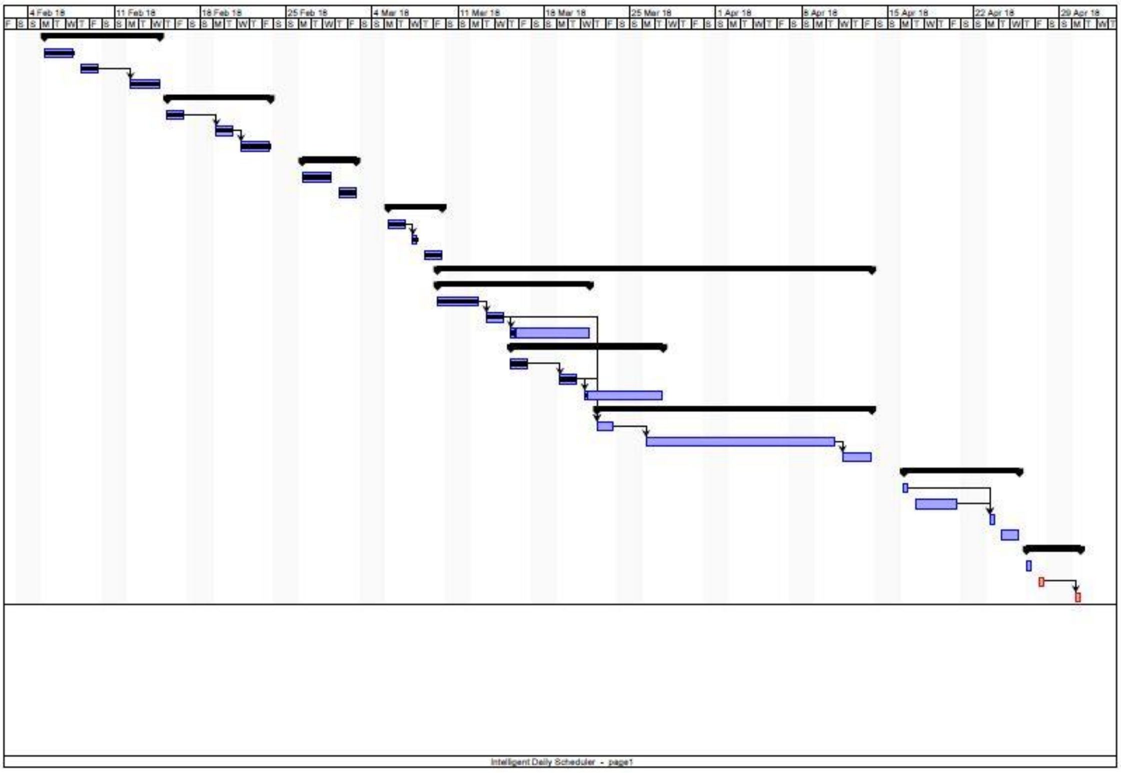


Fig 4.2 Gantt Chart

4.2 Risk Identification

4.2.1 Risk of a highly uncertain day schedule of a person

Probability	Medium
Impact	High
Exposure	High

Mitigation steps: Reinforce the schedule

4.2.2 Risk of overwhelming the server

Probability	Medium
Impact	Medium
Exposure	Medium

Mitigation steps: Enforce load balancing by spreading the requests across the day

5. Software Requirement Specifications

5.1 Product Overview

The product (mobile app) developed through this project focuses on efficiently scheduling people& day-to- day activities. The project is divided into two parts in which the first part focuses on estimating the leisure/free time of a person which is calculated on the basis of information collected via the app. The second part is enabling the app to schedule the daily activities of the user by enforcing load balancing by employing an appropriate scheduling algorithm.

We predict the leisure/free time of a person by employing deep learning techniques as compared to other traditionally used machine learning techniques. We used feed forward deep neural network model for effectively predict leisure time as they have memory and take into consideration the consistency of a person schedule. This project will be using pre-processed data set (Activities of Daily Living (ADLs) Recognition Using Binary Sensors Data Set) for testing purpose. Later we use data collected from user through android application interface. We have to find a scheduling algorithm which is best suited to allocate all tasks in free time without over scheduling tasks/jobs on single day.

5.2 External Interface Requirements

5.2.1 User Interfaces

Android App: The android app is a bridge between user and the functionality of the project. This application will be used by users to interact with the server which has the trained neural network model through the http request and response. The users should be able to enter the tasks which they want to complete. This application should also collect data about free time of users and suggest a task when they are free. This application should also provide good interface to keep track of user performance in completing their daily goals.

5.2.2 Hardware Interfaces

Android Device: Android device is a model used to launch the Intelligent scheduler application. User should have a good Smart phone which has a minimum of 1 GB RAM and 8 GB memory with android version should be Android 5.0 (Lollipop) and above.

Server (Google Cloud Server): Server is needed for the project as neural network model should be trained with huge data collected from different users. The user can communicate to the server through the http request and response. Input to the Neural network model is given through server. The model is trained in the server and it returns the predicted values. Server is also used to schedule the tasks according to algorithm.

5.2.3 Software Interfaces

Neural Network Model (TensorFlow): The most important predicting is to find free time of users. This can be done by building neural network model which predicts based on previous data. Tensorflow can be used to build a neural network model. Tensor flow is chosen as it is the most used and is compatible with lot of servers.

Firebase-Real Time Database: Firebase is used for the storage purpose in this project. It is also used because it can be accessed from anywhere and the speed of accessing the data about each user is very fast. The information such as free time of users and the different tasks/jobs is stored here.

5.2.4 Communication Interfaces

HttpClient: The Hypertext Transfer Protocol (HTTP) is an application protocol for distributed, collaborative, and hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web. HTTP functions as a request-response protocol in the client–server computing model. Purpose of HTTPClient is to transmit and receive HTTP messages. All HTTP requests have a request line consisting a method name, a request URI and an HTTP protocol version. The client side of http is concerned with sending http requests via GET or POST methods to the server. HTTP response is a message sent by the server back to the client after having received and interpreted a request message. The obtained response is processed at the client side for application specific purposes.

5.3 Functional Requirements

5.3.1 Functional Requirement-1

Case name	Track User’s free time.
Actors	User, Android device, Database server
Precondition	• Connection should be established between mobile and server should be established.
Main Success scenario	• The application should interact with the user and collects data from him regarding the free time on a daily basis. • Collection of user’s free time in the database.
Exception	• Network problem • Low Storage in Database

5.3.2 Functional Requirement-2

Case name	Notify Users of task.
Actors	User, Android device.
Precondition	• User should have given few tasks. • NNmodel should have predicted free time.
Main Success scenario	• The user should be informed of task/job he should complete through the app.
Exception	• NN model would not have predicted free time properly • There is no task specified from user

5.3.3 Functional Requirement-3

Case name	Schedule the tasks.
Actors	User, Android device, Server.
Precondition	• Server should have access to tasks/jobs db. • Server should have access predicted free time of user. • Server should be active.
Main Success scenario	• Server schedules the jobs according to scheduling algorithm.
Exception	• Server does not have access to db due to poor network.

5.3.4 Functional Requirement-4

Case name	Week statistics
Actors	User, Android device, Server.
Precondition	• Db should be connected to mobile
Main Success scenario	• The application should present the required statistics to user in the form of visuals such as pie charts and graphs indicating the summary of the week.
Exception	• Poor network connection.

5.3.5 Functional Requirement-5

Case name	Load Balancing of tasks
Actors	Android device, Server.
Precondition	• Server should have access to tasks/jobs db. • Server should have access predicted free time of user.
Main Success scenario	• The server should schedule the tasks in such a way that over scheduling of tasks in one day should not happen.There should be load balancing techniques to schedule properly
Exception	• Server does not have access to db due to poor network.

6. Design

6.1 Introduction

The system design can be fairly abstracted to be consisting of two divisions namely the application and server side. This way of division helps to understand the interactions between various components of the system and their inter dependencies.

The application block represents the interaction between the user and the mobile application. It is responsible for collecting data from the user and present him with the required information. The server block represents the interaction between the application and the server. It is responsible for passing the data collected from the application and render it to the vital components ensuring they work in harmony and return the processed information to the application which in turn renders information to the user.

6.2 Architecture Design

The architecture design can be visualized to be composed of components interacting with each other exchanging data. The flow can be best represented by a diagram.

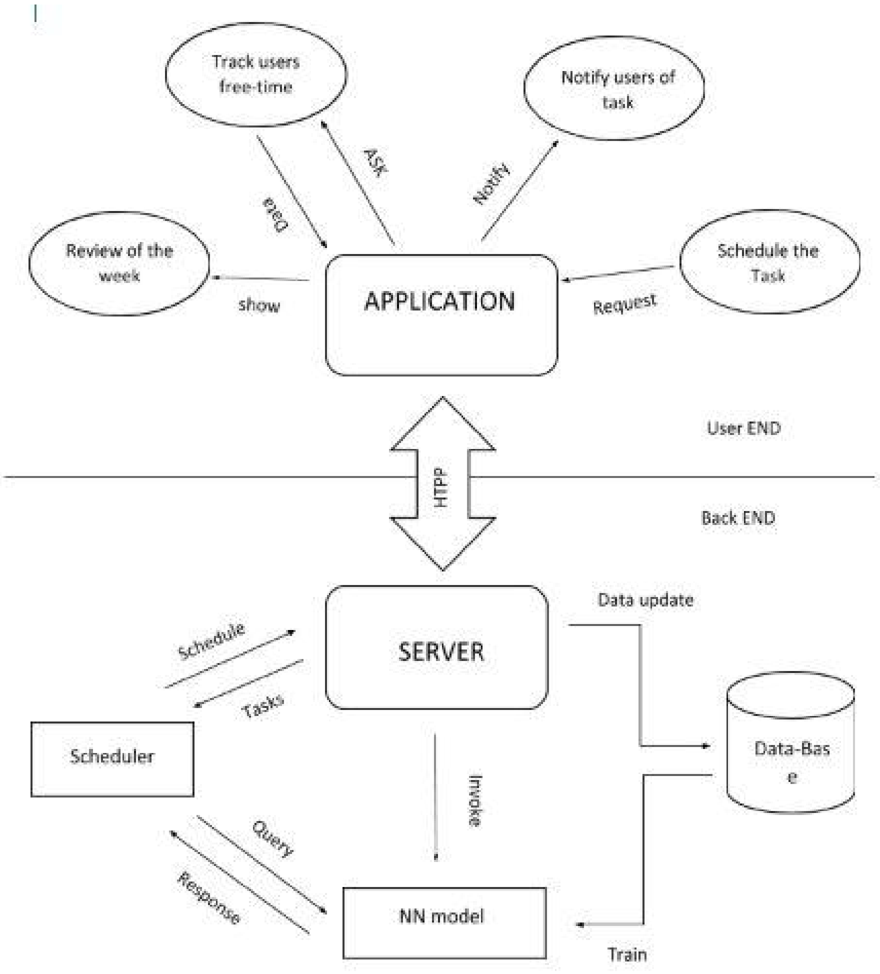


Fig 6.1 System design

6.3 Graphical User Interface

User interface forms a vital part of any application. As smartphones have become ubiquitous, an android app would make a convenient choice for our application. The user interface provided by an android application is intuitive and easy to use. All the major tasks to be performed are divided into android activities and fragments.

All the activities are made navigable by the use of android menus. They provide buttons that map to the corresponding activities. The major activities include the ones to create new tasks, display the schedule to the user and to manage their tasks. Recycler views are made use of because they are efficient in displaying list items compared to the traditional list views.

Activity that displays the schedule is further divided into fragments which map to the days of the week. Fragments are lightweight activities that have a similar life cycle of a typical android activity but run independently thus avoiding the need for the entire frame to be refreshed when they are loaded.

6.4 Class diagrams and classes

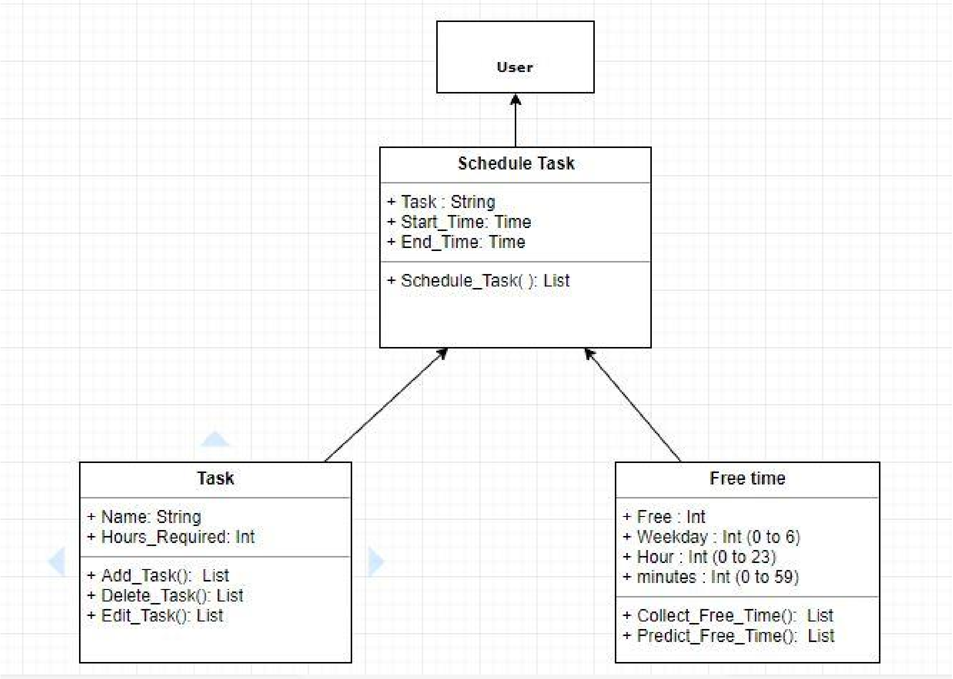


Fig 6.2 Class diagram

There are three main classes in this project which are shown here

- *Task Class:* This class Indicates Tasks to be scheduled. This has two attributes one for storing the name and other for indicating the total no of hours. The methods of the class is Add_Task(), Delete_Task() and End_Task().
- *Free Time class:* This class is used to store whether user is free or not in 15 minutes interval. The attributes of this class are Free (indicating whether user is free or not in that interval) , Weekday (indicating which day of the week), Hour (indicating which hour) and minutes (indicating which minutes). The methods of the class are Collect_Free_Time() and Predict_Free__Time.

- *Schedule_Task class:* This class contains all the tasks scheduled in time slots. The attributes of this class are Task (Task name), Start_Time and End_Time. The methods of the class are Schedule_Task() which is used to schedule tasks in the free intervals.

6.5 Sequence Diagram

The sequence diagram is represented as follows

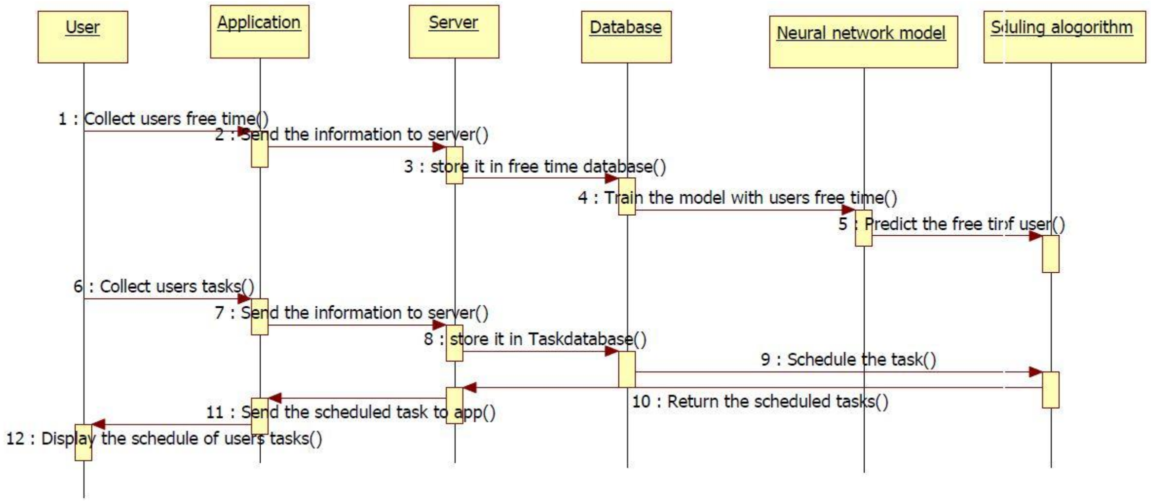

User
Application
Server
Database
Neural network model
Sduling alogorithm
1 : Collect users free time()
2 : Send the information to server()
3 : store it in free time database()
4 : Train the model with users free time()
5 : Predict the free tinf user()
6 : Collect users tasks()
7 : Send the information to server()
8 : store it in Taskdatabase()
9 : Schedule the task()
10 : Return the scheduled tasks()
11 : Send the scheduled task to app()
12 : Display the schedule of users tasks()

Fig 6.2 Sequence diagram

6.5 Data Flow Diagram

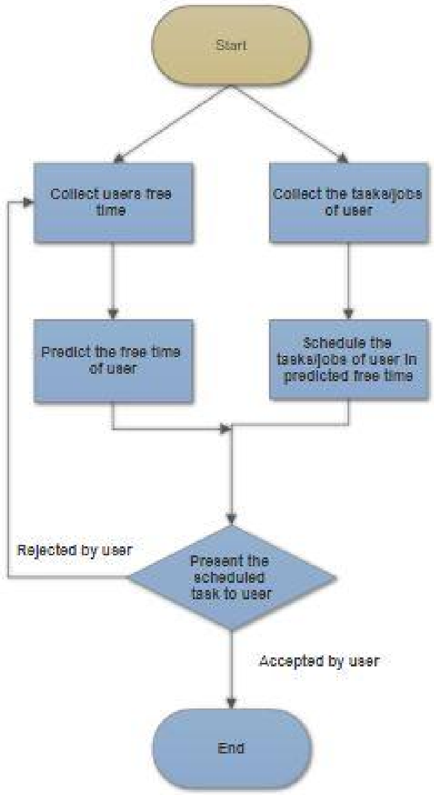

Start
Collect users free time
Collect the tasks/jobs of user
Predict the free time of user
Schedule the tasks/jobs of user in predicted free time
Rejected by user
Present the scheduled task to user
Accepted by user
End

Fig 6.2 Data Flow diagram

6.7 Conclusion

The overall view of the system has been provided and an abstraction of how the components are interacting with each other can be ascertained. The architecture diagram, sequence diagram and the data flow diagram show the flow of data and interactions among the components which form the vital part for the coding process. A proper understanding of the components helps in construction of modules.

7. Implementation

7.1 Tools Introduction

7.1.1 Tensor flow

TensorFlow is an open source software library for high performance numerical computation.[6] Its flexible architecture allows easy deployment of computation across a variety of platforms (CPUs, GPUs, TPUs), and from desktops to clusters of servers to mobile and edge devices. Originally developed by researchers and engineers from the Google Brain team within Google's AI organization, it comes with strong support for machine learning and deep learning and the flexible numerical computation core is used across many other scientific domains. Using TensorFlow, we can build and deploy a neural network model.

7.1.2 Pandas

Pandas is an open source, BSD-licensed library providing high-performance, easy- to-use data structures and data analysis tools for the Python programming language[7]. Using Pandas, we can easily load data, pre-process it for the machine learning models and do operations on data.

7.1.3 Google Cloud

Google Cloud infrastructure is engineered to handle the most data-intensive work, giving flexibility to scale quickly, while still maintaining admin control. GCP frees you from the overhead of managing infrastructure, provisioning servers, and configuring networks. Google data analytics and easy-to-use machine learning services help in capture, process, and analyze data within a single platform. GCP also provide Rs 20,000 credit which providing better overall value.

7.1.4 Android Studio

Android Studio is Android's official IDE. It is purpose-built for Android to accelerate development process and help build the high-quality apps for every Android device. Android Studio offers build automation, dependency management, and customizable build configurations. We can configure projects to include local and hosted libraries and define build variants that include different code and resources, and apply different code shrinking and app signing configurations.

7.1.5 Firebase Realtime Database

The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON and synchronized in realtime to every connected client. Instead of typical HTTP requests, it uses data synchronization and every time data changes the connected devices are updated.The database can be accessed directly from a mobile device or web browser and there’s no need for an application server. Security and data validation are available through the Firebase Realtime Database Security Rules, expression-based rules that are executed when data is read or written. The Realtime Database is a NoSQL database and as such has different optimizations and functionality compared to a relational database. The Realtime Database API is designed to only allow operations that can be executed quickly. This enables building a great realtime experience that can serve users without compromising on responsiveness.

7.2 Technology Introduction

7.2.1 Deep Neural Network (DNN)

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurones) working in unison to solve specific problems. ANNs, like people, learn by example. An ANN is configured for a specific application, such as pattern recognition or data classification, through a learning process. Deep Neural Network (DNN) is a part ANN but with multilayer network [8].

7.2.2 Cloud Computing

Simply put, cloud computing is the delivery of computing services—servers, storage, databases, networking, software, analytics and more—over the Internet ("the cloud"). The advantages of Cloud Computing are

- Flexibility: Cloud-based services are ideal for growing or fluctuating bandwidth demands. If there is increase in demand it's easy to scale up your cloud capacity, drawing on the service's remote servers. Likewise, if there is a need to scale down again, the flexibility is baked into the service.

- Work from anywhere: With cloud computing, if there is a good internet connection we can connect anywhere to cloud. And with most serious cloud services offering mobile apps, there is no restriction on device user has got to hand.

- Increase speed and agility: In a cloud computing environment, new IT resources are only ever a click away, which means you reduce the time it takes to make those resources available to developers from weeks to just minutes. This results in a dramatic increase in agility for the organization, since the cost and time it takes to experiment and develop is significantly lower.

7.2.3 Android

Android is a mobile operating system developed by Google, based on a modified version of the Linux kernel and other open source software and designed primarily for touchscreen mobile devices such as smartphones and tablets. It has gained the popularity of being the most used OS for smartphones. Android source code is free for anyone to download, customize, and distribute. This lets manufacturers build mobile devices at lower costs, giving people around the globe access to mobile technology that was previously out of reach.

7.3 Overall view of the project in terms of implementation

7.3.1 Application

This block represents the interaction between the user and the mobile application. It is responsible for collecting data from the user and present him with the required information. This task is carried out as follows.

Track user's free time: The application interacts with the user and collects data from him regarding the free time on a daily basis. This forms a major part of data extraction and helps the application distinguish users.

Notify users of the task: The application provides all the necessary notifications to the user regarding the time scheduled for his activities.

Schedule the tasks: Post the free time analysis of the users, they can ask the application to schedule their tasks in an appropriate manner. The application takes in the tasks given by the user and schedules them appropriately.

7.3.2 Server

This block represents the interaction between the application and the server. It is responsible for passing the data collected from the application and render it to the vital components ensuring they work in harmony and return the processed information to the application which in turn renders information to the user.

Database: The data received form the application is redirected to the database by the server. Database is responsible for storing all the data ensuring isolation between data of various users. It forms a vital component as it has to store all the data in an organised form and deliver it to various other components in the format they require.

NN model: The neural network model is the key section which helps to personalise the application. The server dictates timely training of the model which is done by extracting the required amount of data from the database and modelling. The models represent the users state.

Scheduler: When the server receives a request from the application, the scheduler is invoked. It is the responsibility of the scheduler to interact with the trained neural network model and based on the information of the user's state received, the scheduler decides the user’s schedule.

7.4 Explanation of Algorithm and how it is been implemented

We have taken into consideration of factors such as minimum task juggling and load balancing to design a scheduling algorithm. The goal of our algorithm is to schedule the n tasks given by users in the free time predicted. At the first step of algorithm T Total Time Required to Complete all the Tasks. Given t<n> Time Required to Complete a Task we can calculate the T Total Time Required to Complete all the Tasks. We will get f<n> Free Time for the Day from the predicted free time slots. We can calculate F Total Free Time in a Week by summation of free time for seven days. The R Total Rest Time is calculated by subtracting F Total Free Time in a Week and T Total Time Required to Complete all the Tasks. When R is negative there is no way to schedule all tasks in given Predicted free time. Therefore, we send an error message to the user. If R is positive, we divide the R Total Rest time evenly to all 7 days to get r Rest Time for individual days. This achieves load Balancing as it ensures that each day as sufficient rest time and no day is overcrowded with tasks.

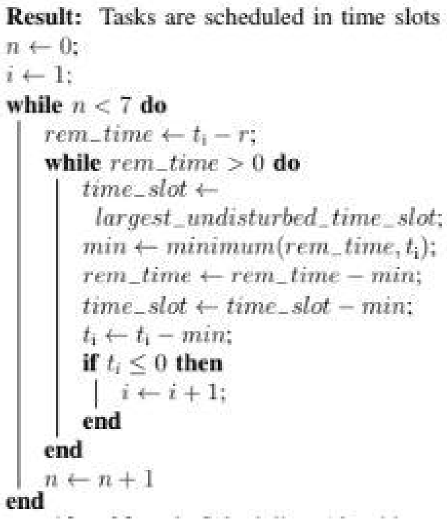
$$T = \sum_{i=0}^{n} t < i >$$

$$F = \sum_{i=0}^{n} f < i >$$

$$R = F - T$$

$$r = \frac{R}{7}$$

Next, we need to allocate time slots for task. To do this we find Remaining time ‘rem_time’, which is calculated form subtracting t<i> with r for all seven days. if rem is greater than 0 indicates favourable situation. To minimise the Task juggling we have to find the largest undisturbed free time slots ‘time_slot’. Each cycle of iteration we find minimum between ‘time_slot’ and t<i>. The task i is scheduled in this time interval. This minimum is subtracted with rem_time, time_slot and t<i>. if t<i> is zero next task is chosen (i.e. i = i+1). We find out find the next largest undisturbed time interval and repeat the process until rem time is zero. This entire process is done for all seven days.



```
Result: Tasks are scheduled in time slots
$n \leftarrow 0$;
$i \leftarrow 1$;
while $n < 7$ do
    $rem\_time \leftarrow t_i - r$;
    while $rem\_time > 0$ do
        $time\_slot \leftarrow$
          $largest\_undisturbed\_time\_slot$;
        $min \leftarrow minimum(rem\_time, t_i)$;
        $rem\_time \leftarrow rem\_time - min$;
        $time\_slot \leftarrow time\_slot - min$;
        $t_i \leftarrow t_i - min$;
        if $t_i \leq 0$ then
            $i \leftarrow i + 1$;
        end
    end
    $n \leftarrow n + 1$
end
```

7.5 Information about the implementation of Modules

A typical system model for intelligent scheduler consists of three modules

- Data collection module
- Free time prediction module
- Task scheduler module.

7.5.1 Data collection module

As smartphones have become ubiquitous, an android application would make a convenient choice for collecting data from users. The initial phase of data collection is carried out once the user installs the application. Here the information regarding the user's rough estimate of leisure time in each weekday is gathered. This accounts for the first degree of personalisation. Following this phase, the user has the flexibility to feed in his/her list of activities to be performed as per the requirement.

The second phase of data collection is rather indirect and dynamic[8]. It is also notable that the frequency of data collection in this phase depends upon the exactness of the data collected in the first phase and the variation of leisure time presented by the user. The system is said to be in a state which represents the user's behaviour. Each time a list is fed in by the user, the system takes the action based on the state and schedules the activities. Once the schedule is presented to the user and if it is accepted, the state remains unchanged and data is not collected. In the other scenario, where the user does not accept the schedule represents a case of variation. Here an indirect data collection is carried out without the knowledge of the user where the system state is modified to account for the variation in user's behaviour.

7.5.2 Free time prediction module

Once we have free time from the user we have to predict when user is free for the next week. To do the above task we have divided dataset into 15 minutes interval. It then becomes a pattern classification problem whether the user is free or not in 15 minutes interval. To predict pattern in user's daily schedule we have made use of Deep Learning Neural Network model. As discussed in [6], the main difficulty of pattern classification algorithms is that complexity increases as we increase the dimensionality of data. With a robust deep learning it's possible to train on large set of observation and predict the pattern in user's daily life.

For TensorFlow DNN (Deep neural network) model Construction we used three-layer back propagation network. The number of nodes in hidden layer is 4 and 2. The base features columns for DNN model are weekday, hour and minute. Crossed features columns can make interaction between base features more effectively. The Crossed Feature columns for DNN are weekday_hour, hour_minute and weekday_hour_minute. For the dataset the DNN classifiers provides 78.2% accuracy. This is good enough accuracy considering the variability of user's schedule and only having less than 13 days of data.

We use the DNN classifier developed above to predict real time users schedule for the next week. The data collected by the user through his phone increases every week. Therefore, the accuracy of the prediction also increases.

7.5.3 Task scheduler module

Staying organised and planning a day ahead can increase individual's efficiency. This is the reason why a good scheduling algorithm can be highly impactful in one's life. But the qualities of scheduling algorithm is uncertain. One of the features which to be followed is to avoid multitasking .Contrary to the belief, multi-tasking can decrease efficiency of task completion. An individual may forget the status of the work he was doing before, this affects the speed and quality of the individuals output. Therefore, it is better to complete one job before moving on to other. Another feature to consider is balancing of tasks which is also known as Load balancing. Taking proper rest is crucial ingredient in time management. In our approach we take into consideration of above factors (i.e. minimum task juggling and load balancing) to design a scheduling algorithm. The goal of our algorithm is to schedule the n tasks given by users in the free time predicted.

7.6 Conclusion

The application has been carefully subdivided into modules such that isolation between them is ensured. Changes made to one module do not significantly affect the functioning of the other modules. The application has been designed in manner that it is light weight and does not burden the user. Both the neural network model and the scheduling algorithm run on the server ensuring the user a hassle free experience. Overall the user is provided with a fluid application with relatively low response times.

8. Testing

8.1 Introduction

Software testing is an activity to check whether the actual results match the expected results and to ensure that the software system is defect free. It involves execution of a software component or system component to evaluate one or more properties of interest. Application testing is a process through which the functionality, usability, and consistency of the entire application are tested. Another major component of software testing is performance testing. This subdomain encompasses various aspects like latency, response times and optimizations.

8.2 Testing tools and environment

8.2.1 Frontend testing

Mobile application testing encompasses the functionality testing of the applications that run on mobile applications. Mobile application testing is far more challenging compared to desktop application testing owing to the fact that mobile apps have to run on various devices ensuring compatibility and uncluttered UI feel. In case of android there are several versions and mobile devices running android feature varied screen sizes and vendor specific optimizations.

Firebase Test Lab

Firebase Test Lab provides simple yet comprehensive way of performance testing of android apps. It provides an overall view of the app performance by evaluating certain parameters such as synchronization and latency. Statistics can be obtained regarding CPU usage, memory usage and network transactions. The statistics obtained for a sample run of the application is as follows.

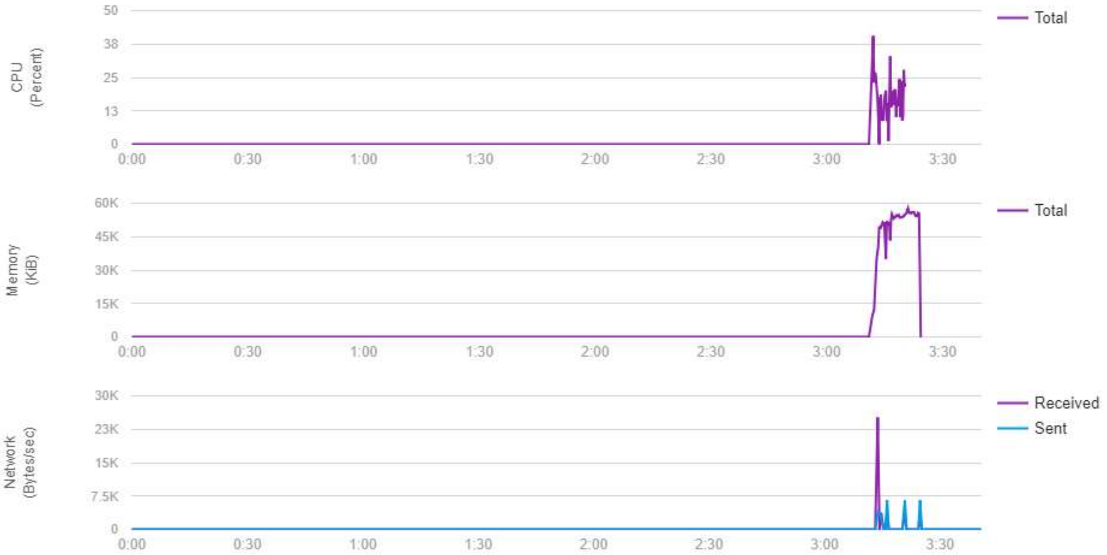
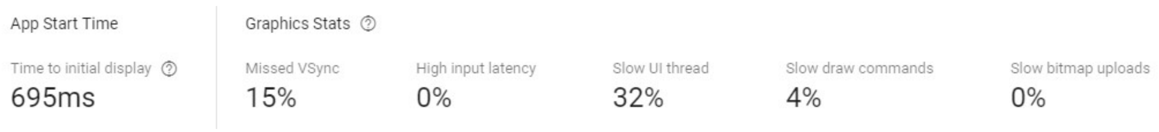


Fig 8.1 CPU usage, memory usage and network transactions statistics

Graphical statistics such as percentage of frame rate drops, missed synchronizations and latencies can also be obtained. This gives an insight regarding the overall response time of the application and indicates how fluid the application is.



App Start Time	Graphics Stats				
Time to initial display	Missed VSync	High input latency	Slow UI thread	Slow draw commands	Slow bitmap uploads
695ms	15%	0%	32%	4%	0%

Fig 8.2 Overall application performance statistics

8.2.2 Backend testing

tf.confusion_matrix: Tensor flow has a method tf.confusion_matrix which computes the confusion matrix from predictions and labels. The matrix columns represent the prediction labels and the rows represent the real labels. The confusion matrix is always a 2-D array of shape [n, n], where n is the number of valid labels for a given classification task. Both prediction and labels must be 1-D arrays of the same shape in order for this function to work.

sklearn.metrics.roc_curve and sklearn.metrics.auc: SkLearn has a method sklearn.metrics.roc_curve to compute Receiver operating characteristic (ROC) It returns fpr and tpr which is increasing false positive rates and true positive rate respectively. This is used to represent specificity (False Positive Rate) and their sensitivity (True Positive Rate). sklearn.metrics.auc is used to compute Area Under the Curve (AUC) using the trapezoidal rule. This is a general function, given points on a curve it returns a float value auc (Area under Curve).

8.3 Test cases

8.3.1 Neural Network Model:

For Testing purpose, we choose Activities of Daily Living (ADLs) Recognition Using Binary Sensors Data Set. This dataset is comprising information of daily activities of user based on sensors in his home for a total of 13 days. We have considered ‘Spare_Time/TV’ as free time. This data set is divided in 70-30% training and testing set. We will focus on the following testing analysis methods for the DNN model since it is binary classification problems:

Confusion Matrix:

The confusion matrix is an mxm, where m is the number of classes to be predicted. For binary classification problems, the number of classes is 2, thus the confusion matrix will have 2 rows and columns. The rows of the confusion matrix represent the target classes while the columns represent the output classes. the predicted results are compared to the actual values of the target variables. This gives us four possibilities:

- True positives (TP), which are the instances that are positives and are classified as positives.
- False positives (FP), which are the instances that are negatives and are classified as positives.
- False negatives (FN), which are the instances that are positives and are classified as negatives.
- True negatives (TN), which are the instances that are negatives and are classified as negatives

The confusion matrix is

	Predicted positive	Predicted negative
Actual Positive	190	100
Actual negative	36	110

Table 8.1 Confusion matrix without normalization

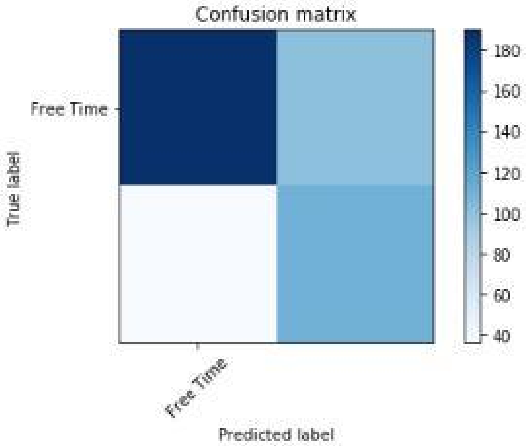


Fig 8.3 Confusion matrix without normalization

Normalised Confusion Matrix by row is

	Predicted positive	Predicted negative
Actual Positive	0.66	0.34
Actual negative	0.25	0.75

Table 8.2 Confusion matrix without normalization

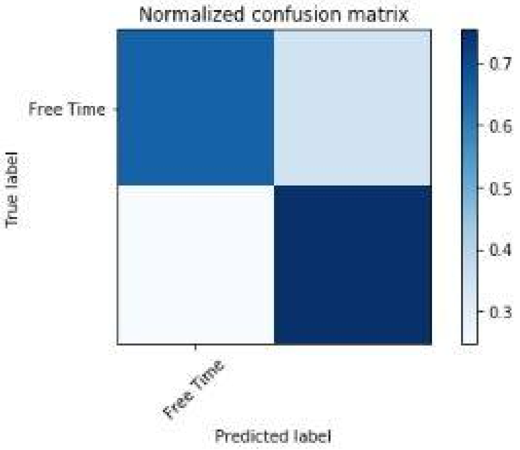


Fig 8.4 Confusion matrix with normalization

ROC curve:

Receiver operating characteristic (ROC) curve is one of the most useful testing methods for binary classification problems since it provides a comprehensive and visually attractive way to summarize accuracy of predictions. By varying the value of the decision threshold between 0 and 1, we obtain a set of different classifiers for which we can calculate their specificity (False Positive Rate) and their sensitivity (True Positive Rate).

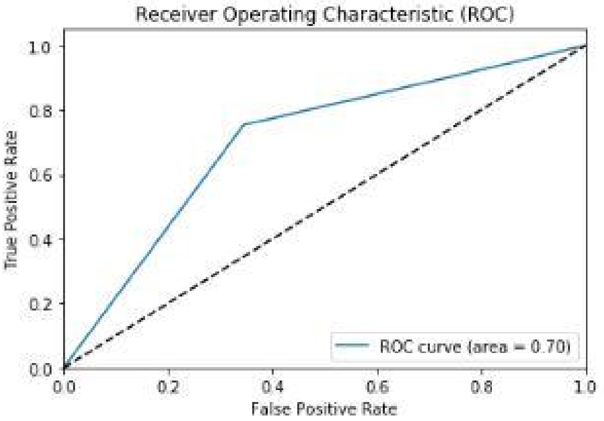


Fig 8.5 ROC Curve

In the case that we had a perfect model, the ROC curve would pass through the upper left corner, which is the point in which the sensitivity and the specificity take the value 1. As a consequence, the closer to the point (0,1) the ROC curve, the better the classifier.

	Value
Area under curve (AUC)	0.7

Table 8.3 ROC Curve

The most important parameter that can be obtained from a ROC curve is the area under the curve (AUC), which is used as a measure of the quality of the classifier. For a perfect model, the area under the curve would be 1. For our example, the AUC is 0.704298, which shows the good performance of our classifier considering the variability of Daily schedule of the Test Subject.

8.3.1 Database Testing:

Objective	Update the tasks of users in the database.
Prerequisites	• Successful authentication • Obtain the correct database reference object
Execution	Push data into the firebase database in the form of ToDolist objects.
Potential bugs	• Failure of authentication • A possible null database reference
Results	The tasks were updated as expected.

9. Conclusion and scope for future work

The main objective of the intelligent scheduler application is to provide a substantial solution to the problem faced by individuals in scheduling their activities. The traits of this application includes an analysis of the leisure time of an individual which tends to show a certain degree of variation. Widespread use of smartphones and their proven utility convinced us to build an android application as the interface. Data collection is done via the app and users have the flexibility of feeding in their list of tasks to be performed according to their convenience. The scheduling algorithm used plans the tasks in such a way that besides personalization the criterion of load balancing is met. Both the neural network model and the scheduling algorithm used remain isolated from the user's phone as they are run on the cloud without burdening the user. This ensures that the application is light and ensures faster response times. All these features provide a customized experience to the users and help boost their efficiency.

The application can be extended to incorporate google calendar. Also statistics regarding the usage time of apps such as Whatsapp, YouTube can be shown to the user. Currently the accuracy of the neural network model is hovering around 75%. This can be improved to around 80-85%.

10. References

1: Misra, Ranjita, and Michelle McKean. "College students' academic stress and its relation to their anxiety, time management, and leisure satisfaction." *American Journal of Health Studies* 16, no. 1 (2000): 41.

2: Jackson, Valerie P. "Time management: a realistic approach." *Journal of the American College of Radiology* 6, no. 6 (2009): 434-436.

3: Coviello, Decio, Andrea Ichino, and Nicola Persico. "Time allocation and task juggling." *American Economic Review* 104, no. 2 (2014): 609-23.

4: Wiers, Vincent CS. "A review of the applicability of OR and AI scheduling techniques in practice." *Omega* 25, no. 2 (1997): 145-153.

5: Baras, Karolina, Luísa Soares, Norberto Paulo, and Regina Barros. "'Smartphine': Supporting students' well-being according to their calendar and mood." In *Computer and Energy Science (SpliTech), International Multidisciplinary Conference on*, pp. 1-7. IEEE, 2016.

6: Abadi, Martín, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin et al. "TensorFlow: A System for Large-Scale Machine Learning." In *OSDI*, vol. 16, pp. 265-283. 2016.

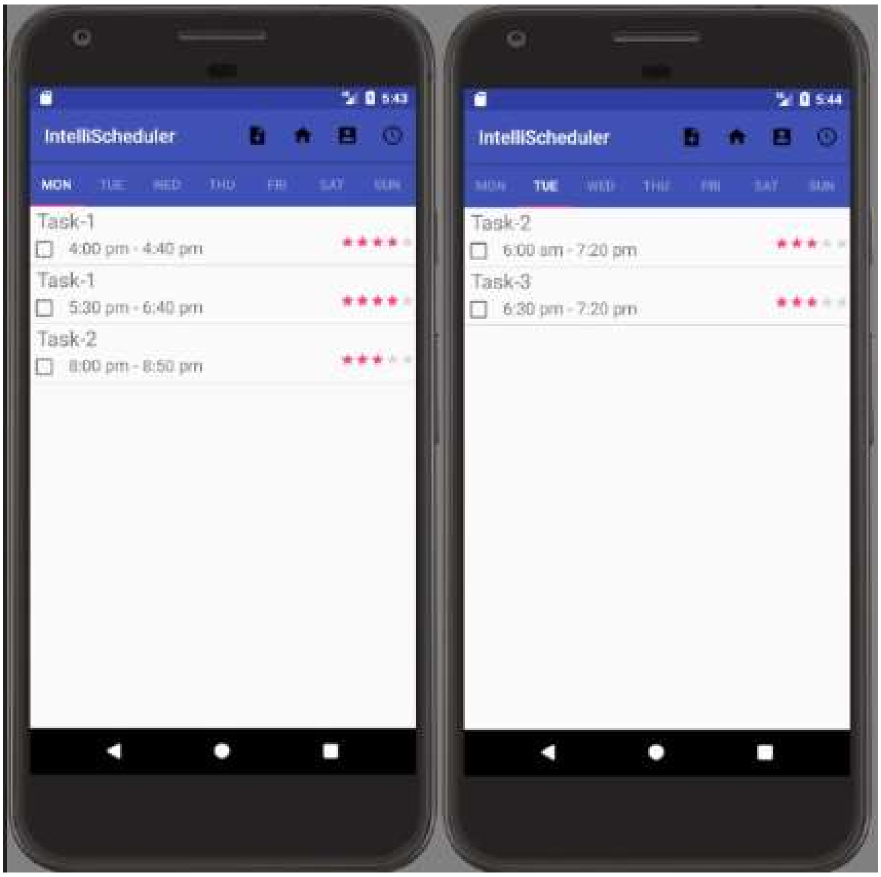
7: McKinney, Wes. "pandas: a foundational Python library for data analysis and statistics." *Python for High Performance and Scientific Computing* (2011): 1-9.

8: Glorot, Xavier, and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks." In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249-256. 2010.

11 Appendix

Screen Snapshots

The following screenshot provides insight regarding the running application.



Task-1
□ 4:00 pm - 4:40 pm ★★★★

Task-1
□ 5:30 pm - 6:40 pm ★★★★

Task-2
□ 8:00 pm - 8:50 pm ★★★

Task-2
□ 6:00 am - 7:20 pm ★★★

Task-3
□ 6:30 pm - 7:20 pm ★★★

Fig 11.1 Running application