

Software Requirement Specifications

of

Selected Projects

carried out in

Software Engineering Lab (CS2906)

Spring 2019

Prof. Debasis Samanta



Department of Computer Science & Engineering
Indian Institute of Technology Kharagpur

Indian Institute of Technology Kharagpur
Department of Computer Science and Engineering
Software Engineering Lab (CS29006)

Spring 2019

Project code	Topic	Page
01	Attendance App -I (web-based solution)	1
02	Attendance App – II (mobile solution)	10
03	Secure Document Transmission (mobile solution)	20
04	Artificial Doctor Intelligence (ADI app)	26
05	Document Browser for Blind	31
06	Hand-written Calculator	36
07	Sharing Tour Expenditure	45
08	Online C Program Evaluation	48
09	Chit-chat	55
10	Document Archival	58
11	Garbage Collection	62
12	Food supply from Canteen	71
13	Multimedia Editor(Jedi-t)	79
14	Hindi Editor	86
15	Plagiarism Checker	91
16	Clocked PDA	104
17	Data Entry System	107
18	Online doctor appointment system	112
19	Music recommendation system	125
20	Corporate Banking	136
21	Digital Signature	146
22	Advanced Calculator	157
23	Crowd sourcing for public works(VOTE)	164
24	Online Tic-Tac-Toe Game	176
25	Document Browser	190
26	Course Management System	200
27	WordPro Duo	206

Software Requirements Specification

For

Web-Based Attendance App

Version 1.0 approved

Prepared by

Rohit Pathak (17CS30029)

Nilesh Mandal (17CS10031)

14.02.2019

Table of Contents

Table of Contents	ii
Revision History	ii
1. Introduction	3
1.1 Purpose 3	
1.2 Addressing the Need 3	
1.3 Prospective Users 3	
1.4 Challenges to Overcome 4	
2. Project Planning	4
3. Functional Requirements	5
3.1 User Interfaces 5	
3.2 Hardware Interfaces 5	
3.3 Software Interfaces 5	
4. Nonfunctional Requirements	5
4.1 Performance Requirements 5	
4.2 Safety Requirements 6	
4.3 Security Requirements 6	
5. Software Environment to be Used	6
5.1 Front End Development 6	
5.2 Back End Development 7	
5.3 Database Management System 8	
6. Software Publicity and Cost	8

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

Schools, Colleges, Offices or any such institutions/workplaces need to keep a track of their students/employees for an enhanced performance and smooth running of the institution. Taking attendance is a very trivial yet vital task in numerous institutions. Thus, the need for an attendance app arises due to:

1. **Time Management:** Nowadays, time is short and tasks are many, and hence most institutions cannot afford to waste too much time on attendance. For example, a teacher cannot afford to waste 15 minutes of his 45 minute class during roll call. Thus, it is essential to develop an efficient and fast way of taking attendance.
2. **Keeping a track of employees:** In offices, factories, etc, attendances can help to keep a track of the employees. The attendance can determine which employee is regular and which employee is not.
3. **Inefficiency of Manual Attendance:** In schools, colleges, etc manual attendances are very inefficient and unreliable. There are chances of missed attendance, fake attendance, etc. An attendance app can handle these problems much better.

1.2 Addressing the Need

A few ways in which the need can be addressed are:

1. **Time Management:** The attendance app will use a web-based interface where one must log in using a user-name and password, and then submit his or her attendance. This is a very fast and smooth process.
2. **Keeping a track of work:** The attendance app will keep a track of the attendances of all the employees. It will maintain records about the number of hours attended and generate lists by sorting the employees according to their regularity and work hours.
3. **Inefficiency of Manual Attendance:** The attendance app will require a user-name and password. More personal data can be involved which can reduce the chance of fake attendance. Missed/incorrect attendance will not take place as students are themselves logging into the system and marking their attendance.

1.3 Prospective Users

The prospective users of this attendance app will be school and college students, office employees, factory workers, hospital staff, etc as the attendance markers and by teachers, professors, CEO's, etc as the record keepers.

1.4 Challenges to Overcome

The main issues which we might face while developing the app are:

- Making the user interface simple and easy to use.
- Building a system/method to determine whether the student is actually present at the workplace or is logging in from some other place.
- Maintaining the integrity of the attendance marked,i.e. whether the student himself/herself has logged in to the system or is it someone else who is doing it.
- Designing the web page to ensure easy,fast and yet efficient way of marking the attendance. Thus, the page should be lightweight and the Data Base Management, sorting and other processes should be carried out using efficient algorithms.

2. Project Planning

Week 1 & 2: We will try to learn the concepts required for the project. The project requires a good knowledge of Front End Development tools like HTML,CSS,JavaScript,ReactJS,etc and also a good working knowledge of Java and a Database Management System(MySQL,Oracle,etc).

Week 3: We will try to design the backend for the project using Java. Many actions like Log in, Sign in,Mark attendance will require codes at the backend to process correctly.

Week 4: The designing of the frontend will start. This step will determine how our app will look on visiting the site.Various user interaction buttons and links will have to be integrated in the Web App.The concept of GUI-based programming will be required.

Week 5: The very basic 1st version with minimal features will be completed so that testing and debugging can begin. The most essential functions will be tested before proceeding further.

Week 6 & 7: The app will be updated and more features will be added. Detailed debugging will be done and the app will be improved as much as it can be. Finally when the app will have all the required features and will work satisfactorily, we will deploy the app either on a Cloud based platform or a Local Server.

3. Functional Requirements

3.1 User Interfaces

- 1) GUI along with meaningful Frames and buttons.
- 2) Reports are generated as per requirements.
- 3) User Registration

- 4) Adding new contacts/users.
- 5) Taking attendance.
- 6) Signing in by user.

3.2 Software Interfaces

- 1) Front End - Javascript, HTML and CSS
- 2) Back End - Java
- 3) Random code generating software as a unique code is generated every time attendance is taken.
- 4) Time Tracking (Example - Time Trex) as code is available for a short amount of time.
- 5) Captcha for preventing machine/robot usage
- 6) Integrity Checker

3.3 Hardware Interfaces

- 1) Hardware environment - Dual Core 2nd generation
- 2) System Configuration - RAM - 512 MB HDD 80 GB
- 3) Operating System - Windows XP/Vista/7/8/10/Linux
- 4) Student/Employee Database

4. Nonfunctional Requirements

4.1 Performance Requirements

We need to make sure that the Web Based Application is lightweight so as to ensure fast operation. Each operation should take place smoothly with minimum delay. Each click should have a maximum of 2s delay. The Database management algorithms should be fast so that marking the attendance does not take much time. This is particularly important as there will be a timespan for marking the attendance.

4.2 Security Requirements

One of the biggest threats to security of the system is that it may get logged in to by a code or a bot. To prevent this kind of breach it is essential to include a captcha for verification. Moreover the security system should be full of authenticity checks to provide multi level protection.

4.3 Updates and Patches

The system should be quite flexible to allow for updates and patches. The software development process requires a series of changes to develop the required software with

minimum flaws.

5. Software Environment to be Used

5.1 Front End Development

Mostly Front End Development includes using HTML, CSS, JavaScript, J query, AngularJS (Not everything necessarily). Of all the software tools, the three most used together or separately are HTML, CSS and JavaScript which also will be used by us.

HTML - Hypertext Markup Language (HTML) is the standard markup language for creating web pages and web applications. With Cascading Style Sheets (CSS) and JavaScript, it forms a triad of cornerstone technologies for the World Wide Web. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. HTML can embed programs written in a scripting language such as JavaScript, which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content.

Thus, we will use HTML for the purpose of creating, structuring and appearance of web pages.

CSS - Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content. Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

JavaScript - JavaScript is a high-level, interpreted programming language that conforms to the ECMAScript specification. It is a language that is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm. JavaScript enables interactive web pages and thus is an essential part of web applications. The vast majority of websites use it.

As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative (including object-oriented and prototype-based) programming styles. It has APIs for working with text, arrays, dates, regular expressions, and the DOM, but the language itself does not include any I/O, such as networking, storage, or graphics facilities, relying for these upon the host environment in which it is embedded.

Initially only implemented client-side in web browsers, JavaScript engines are now

embedded in many other types of host software, including server-side in web servers and databases, and in non-web programs such as word processors and PDF software, and in runtime environments that make JavaScript available for writing mobile and desktop applications, including desktop widgets.

Although there are similarities between JavaScript and Java, including language name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design; JavaScript was influenced by programming languages such as Self and Scheme.

5.2 Back End Development

Back End Development mostly includes PHP, Java or any other server side scripting languages. For our project, Java is the language which will be used.

Java - Java is a general-purpose computer-programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture. The language derives much of its original features from SmallTalk, with a syntax similar to C and C++, but it has fewer low-level facilities than either of them.

Why Java -

1. SCALABILITY - Java is highly scalable. Take the case of Java EE. Assuming you have done the right planning and used the right kind of application server, the Java EE can transparently cluster instances. It also allows multiple instances to serve requests. When processing or Input-Output (IO) increases, you can easily add resources, and redistribute the load. Separation of concerns makes this transparent to the app. Java components are easily available, making scaling of large web apps easy. The language is flexible, and you need to do less invasive coding to improve scalability.

2. CROSS-PLATFORM USAGE - You can execute a compiled Java program on all platforms that have a corresponding JVM. This effectively includes all major platforms, for e.g. Windows, Mac OS, and Linux.

You first write your Java program in ".java" file. Subsequently, you compile it using the Eclipse IDE or 'javac', and thereby you create your ".class" files. While it isn't mandatory, you can also bundle your ".class" file into a ".jar" file, i.e. an executable. You can now distribute your ".jar" file to Windows, Mac OS, and Linux, and run it there.

3. POWERFUL MEMORY MANAGEMENT - Java's automatic memory management is a significant advantage. I will describe it briefly here to show how it improves the effectiveness and speed of web apps.

4. MULTI-THREADING - Multi-threading allows multiple users to run one application program simultaneously for their individual tasks. Multi-threading isn't a new concept, for e.g. IBM Customer Information Control System (CICS) has championed it since long in the centralized computing environment. Responsive servers, Quick application response, run multiple operations simultaneously, Lower development time, simplified programming, and reduced application maintenance costs are some of the other advantages.

Few others - Java Security advantages, rich ecosystem and ease of language are some of

the other reasons.

5.3 Database Management System

MySQL - MySQL is an open source relational database management system (RDBMS). MySQL is a component of the LAMP web application software stack (and others), which is an acronym for Linux, Apache, MySQL, Perl/PHP/Python. MySQL is used by many database-driven web applications, including Drupal, Joomla, phpBB, and WordPress. MySQL is also used by many popular websites, including Google (though not for searches), Facebook, Twitter, Flickr, and YouTube.

6. Software Publicity and Cost

Some of the attractive Features of our software:

1. **Enhanced Security:** By incorporating multiple integrity checks by Sign in, Log in(by username and password) and captcha checks, this software is highly secure.
2. **Simple and Easy to Handle:** The interface of the system is really simple and straightforward to enhance easy use and less confusion.
3. **Fast and Efficient:** The software is quite efficient and puts less load on hardware and network. So, this gives it an edge over other attendance apps which take a lot of time in processing.

Some techniques that will help us in marketing our product and drive more traffic to our web app/system :

1. Get active on social media

1. Share your idea on Facebook, Reddit, Instagram, Twitter, LinkedIn, Google+ Stumbleupon, and Forums etc.
2. Invest in Facebook ads.
3. Create a Facebook group and add relevant people to it.
4. Join relevant Google+ groups and talk about your blog
5. Join relevant Linkedin groups.
6. Go on relevant subreddits on Reddit and talk about your project

2. Email Marketing

Send personal Email to an individual that introduce your idea and delivers your Call To Action. We can use softwares like MailChimp for free email listing, creating a newsletter and then sending it out to people.

3. SEO, which is great but time-consuming.

4. **Google Adwords** is great, but also a little expensive and time-consuming.
5. Reach out to influencers through **YouTube, Instagram, and other social media platforms** and let their audience know about our project.

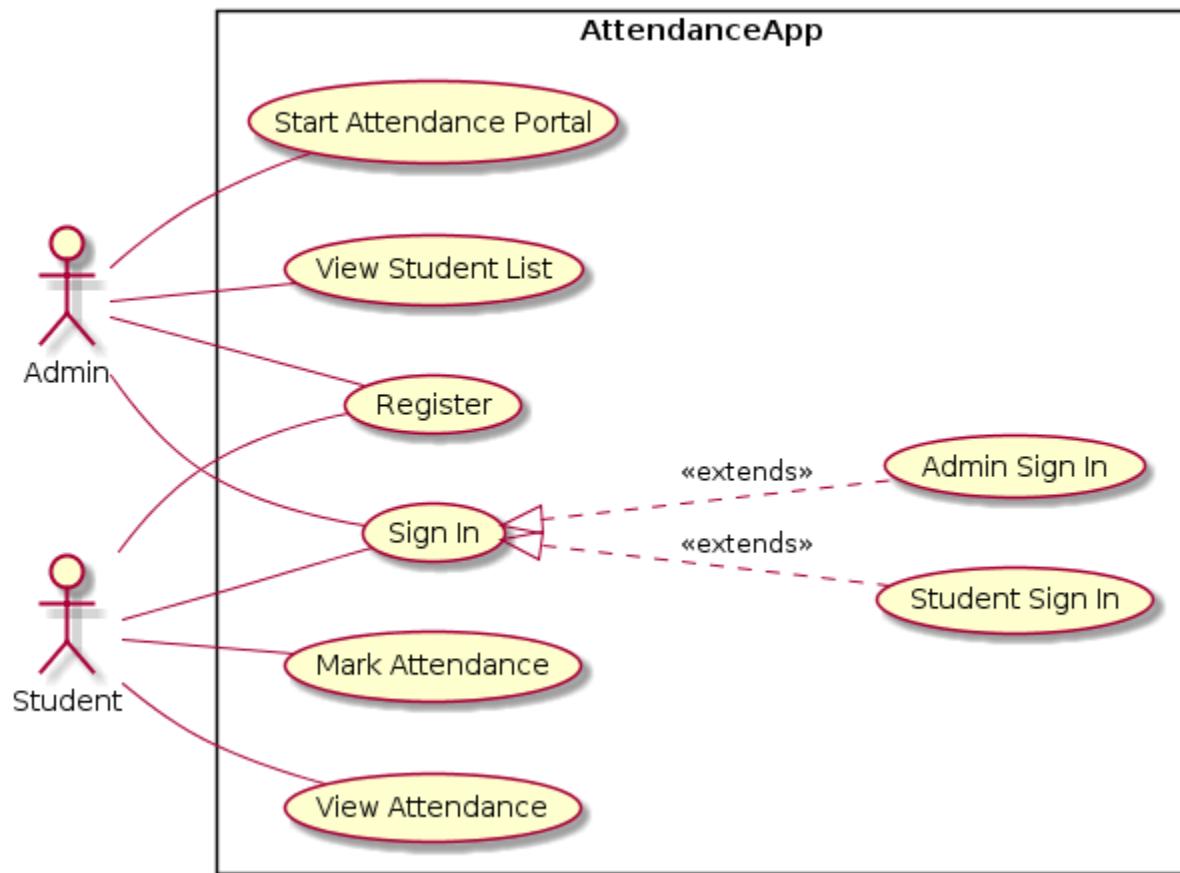
6. Use Landing Pages

Then we just need a high quality dedicated landing page to convert the targeted audience. The goal of the landing page will be to get people to sign up and create our email list.

Cost of the product:

The product will be freely available initially to gain popularity. Then we can earn through ads or make it available to established institutions or individuals at a reasonable price.

USE CASE DIAGRAM:



Code for UML:

```
@startuml
left to right direction
skinparam packageStyle rectangle
actor Admin
actor Student
rectangle AttendanceApp {
    Student -- (Register)
    Admin -- (Register)
    Admin -- (Sign In)
    Student -- (Sign In)

    (Sign In) <|.. (Student Sign In):<<extends>>
    (Sign In) <|.. (Admin Sign In):<<extends>>
    Admin -- (Start Attendance Portal)
    Admin -- (View Student List)

    Student -- (Mark Attendance)
    Student -- (View Attendance)

}
@enduml
```

Software Requirement Specifications for Proxyn't

<Attendance app>

Version 1.0.0

Prepared by

- 1. Karanam Tejendhar (17CS30019)**
- 2. H.O. Sai Varshith (17CS30015)**
- 3. K. Sandeep (17CS10021)**

Indian Institute Of Technology Kharagpur

Date :14/02/2019

Table Of Contents

1. Introduction

- 1.1 Purpose**
- 1.2 Users and Reading Suggestions**
- 1.3 Product scope**

2. Overall Description

- 2.1 Product Perspective.**
- 2.2 Types of Users.**
- 2.3 Supporting Devices.**
- 2.4 Functional Requirements.**

3. Non Functional Requirements

- 3.1 Performance Requirements**
- 3.2 Security Requirements**
- 3.3 Safety Requirements**
- 3.4 Software Quality attributes**
- 3.5 Testability**

4. Hardware and Software Requirements

- 4.1 Hardware Requirements**
- 4.2 Software Requirements**

5. Pricing and Marketing

- 5.1 Marketing**
- 5.2 Pricing**

1. Introduction

1.1 Purpose

The purpose of this document is to present the detailed description about the attendance app. This document explains us about features of the software, the interfaces of the software, what the software will do and the constraints under which it must operate.

1.2 Users and Reading Suggestions

- Teachers ,Professors or or any head of organisation Who wants to take attendance on mobile based application.
- Programmers who are interested in working on the project by further developing it or fix existing bugs.

1.3 Product Scope

- The product scope is very large.
- It is actually very difficult to take attendance for a large number of people simultaneously.
- This application/software makes this difficulty very easy. Actually saves a lot of time .

2.Overall Description

2.1 Product Perspective

This Attendance app software is developed for everyone who wants to save time in taking attendance. It actually works on some unique code which will be decided by the user (who wants to take attendance).

2.2 Types of Users

There are basically three types of users

- Who want to take attendance
- Who give attendance
- Developer or people who update software

2.3 Supporting Devices.

This software is basically used in android.

Not web based software also not for Windows, Mac OS or Linux

2.4 Functional Requirements .

Name Of Function	Input	Output	Justification
login()	Username(), password(), Robot().	Successful or unsuccessfu l	Used to make a registered user do his task
Registration()	Username(),	Successfully	For a new user a

	password(), address(), email(); RollNo;	done or unsuccessfu l Or already registered	person must do registration to do respective task .
robot()	String {Captcha}	-----	People may cheat by taking attendance using robot. Input string is captcha
Username()	String	-----	Name of the user
Password()	String	-----	Password of user. Must be private.
address()	String	-----	Address of the user

3. Non Functional Requirements

3.1 Performance Requirements

To run this software we need android with internet .Mac OS is not supported here. It needs 25 megabytes of RAM. This software can have upto 2000 users.

3.2 Security Requirements

Every user have their respective username and a private password which makes them keep their account safe.

3.3 Safety Requirements

We need to store the data of users, so we need to keep updating the software. There is a bug tracker available where users can report any bugs they have encountered so that the developers can fix it in the next release.

3.4 Software Quality attributes

Attendance app software is very easy to use. Language that we use here in interface is entirely natural language. This user interface is made in very easy natural language and basic symbols.

Who are using this software should basically need to know how to login.

3.5 Testable

We need to get result of every function in user interface in less than 3 sec. This makes user feels good while taking attendance.

4. Hardware and Software Requirements

4.1 Hardware Requirements

- The application only runs on ANDROID devices.
- The device should have at least 500MB of RAM.
- The device should have an active Data Connection.
- The device should not be behind Network proxies or Firewalls.
- The device should have a screen size of at least 4 Inches.
- The app does not work on a wearable.
- The device should have GPS feature enable otherwise the attendance Marked will be reported and administrator takes care of it.
- The Server runs on Linux Machine with a relatively new Processor (newer than i3) and at least 4GB of RAM.
- The Server should have at least 15 GB of disk space to store and Secure the Database.

4.2 Software Requirements

- The app runs on the ANDROID Platform.
- The server runs on the UNIX platform.
- The app needs at least ANDROID JELLYBEAN.
- The server should support JDK 8.
- The server should support MySQL database server.
- The app runs on any device that is supported by the Android SDK for java.
- The server used MySQL JDBC Driver for its Database Management System.
- The server machine should run unix OS that can support multiple parallel Socket connections.
- The server should not block clients with firewall and proxy.

6. Pricing and Marketing

6.1 Marketing

- The app can be marketed as an essential time saver for both students and teachers alike.
- The app can be marketed as eco-friendly as it saves paper.
- The app can also be marketed as fool-proof attendance application as Prevents people from marking false attendances.
- The app also handles attendance of multiple courses. So it can essentially be marketed as a complete attendance solution for any institution.

6.2 Pricing

- The app is available for FREE Download from GITHUB Releases page.
- The server can be used only if it was licensed by the developer.
- The server is priced at Rs. 1000 per Institution per 6 Months.
- The price can be further reduced if the client purchases a licence for a longer period.

Secure document transmission

**P.Madhusudhan Reddy
17CS10033
Kshitiz Sharma
17CS30021**

Secure Document Transmission

The need

For many of us, using some form of online communication to connect with our friends, family and coworkers has become habitual, convenient and second nature. We share a lot of information about ourselves and speak our minds without considering the consequences of having our conversations and data end up in the hands of others, such as government officials, hackers, hosting servers etc. This includes anything from personal photos, sensitive data such as credit card numbers to confidential information about clients within work correspondence .The specific system is developed and designed to beat the attempts of surveillance since no third party can break or decode the data, which is being communicated.

How the need can be addressed?

End-to-End Encryption ensures that all data is encrypted at all times, not just in transit and at rest .Various cryptography techniques like symmetric key algorithms and asymmetric key algorithms help in achieving security goals like authentication, non-repudiation, confidentiality and integrity. In symmetric key cryptography a single key is used for encryption of the data as well as decryption. In asymmetric key cryptography there would be two separate keys. The data which is encrypted using the public key of a user can only be decrypted using the private key of that user and vice-versa.

Prospective users

This service is mainly aimed at any person who is willing to communicate and share his text or any document with other users who might be his/her coworker, friend or family.

Issues/challenges

- **Man-in-the-middle attacks**

End-to-end encryption ensures that data is transferred securely between endpoints. But, rather than try to break the encryption, an eavesdropper may impersonate a message recipient (during key exchange or by substituting his public key for the recipient's), so that messages are encrypted with a key known to the attacker. After decrypting the message, the snoop can then encrypt it with a key that they share with the actual recipient, or their public key in case of asymmetric systems, and send the message on again to avoid detection. This is known as a man-in-the-middle attack.

- **Long term storage and resilience**

If data is stored in a single location or data center then it is subject to data loss, or at least

service interruption whilst that database is offline due to network or hardware failure.

- **Data integrity in changing networking conditions**
- If a server becomes unavailable briefly or for an extended period of time, secondaries and tertiaries must take over immediately to avoid data loss. Devising algorithms that ensure that all servers gossip to each other and agree a consensus on how to assign responsibilities to each server
- **Throttling and rate-limiting is needed to ensure service continuity and avoid service abuse.**

Plan

working on the project starts from feb 26th until April 20th

- front-end (10 days)
 - Home-page, User-page
- Database management(10 days)
 - Storing profile data, chats in a protected form
- Encryption(20 days)
- Debugging and updating(rest of the days)

Functional requirements:

1. User Registration

2. Adding new contacts

3. block or delete contact

4. broadcast messages

5. message status

6 . Delete messages

7 . send message

Non-Functional requirements:

1. Privacy

2. Performance

3. Robustness:

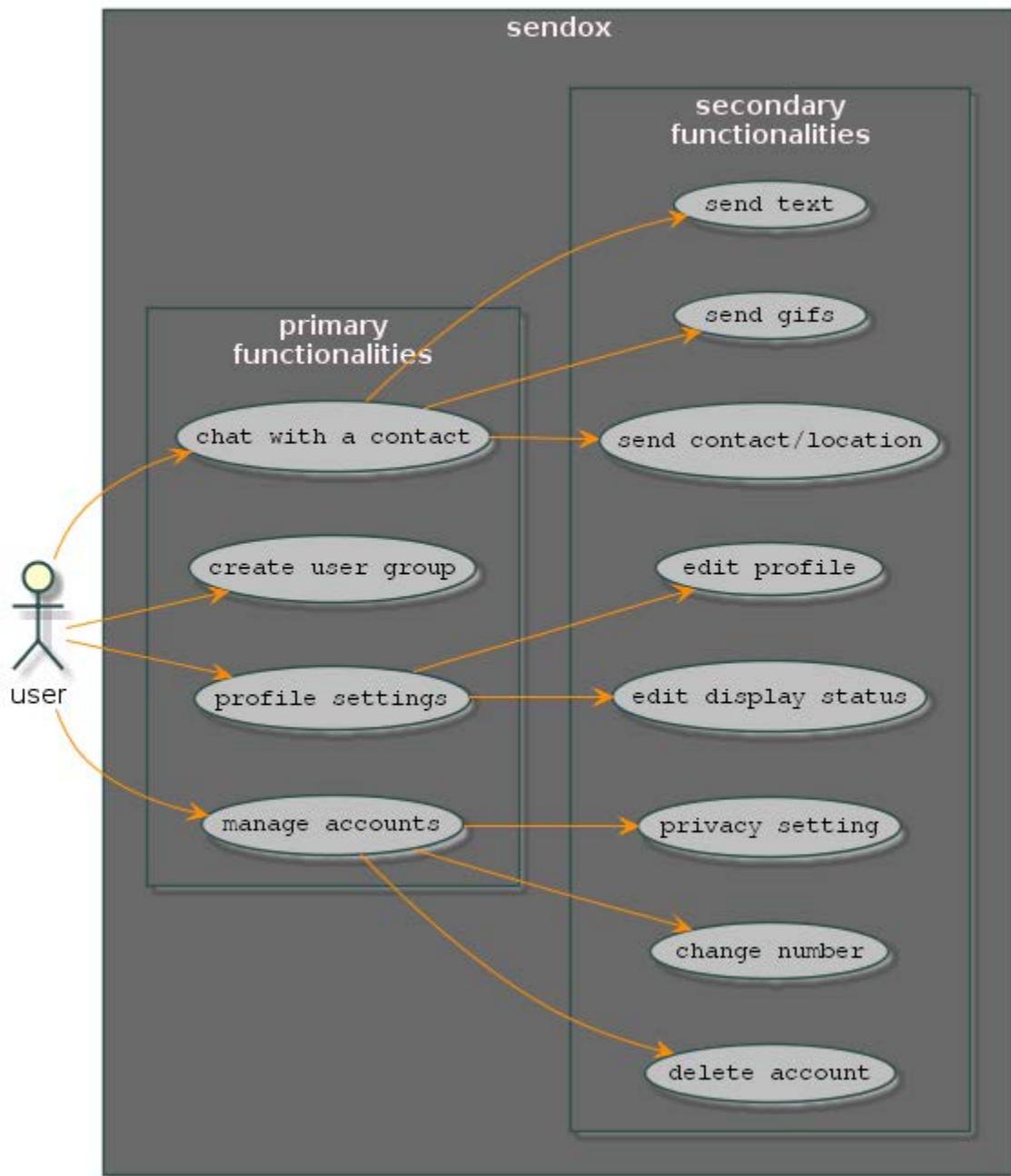
Implementation and tools:

1. Swing for GUI
2. MySQL for data management
3. Networking Protocols:
TCP/IP, HTTPS/HTTP, SMS, SMPP, SMTP etc.
4. Adobe Photoshop for designing

Marketing strategies:

1. Feature app in an official blog
2. Marketing on social medias with adds.
3. App store optimization
4. Getting on an App Review Site
5. Having a good promo video
6. Featuring apps in emails

Use case diagram(UML)



Artificial Doctor Intelligence

Project Overview

The layman do not have enough knowledge about medicines and diseases in order to treat themselves for every illness. In many cases the diseases and illnesses can be cured with medicines that do not require a doctor's consent. Visiting the doctor in such cases is vastly unnecessary and also time consuming and very expensive. Searching for symptoms on Google also doesn't solve the problem as many diseases tend to have similar symptoms. Thus in order to properly diagnose the situation we need the help of an Artificial Doctor. Sometimes we do not recognize that the symptoms we are affected with are quite dangerous, this leads us to not getting a diagnosis until the situation at hand becomes quite severe.

In order to solve these problems we have decided to create an Application that takes into account the symptoms and problems a person is facing. The app will proceed to ask a few questions to figure more about the problem one is facing. We will then make a diagnosis that will help the common person to make an educated decision. We will let the person know how serious his situation is so he can properly know if he can just buy some medicines from the pharmacy or it is necessary for him to get admitted and have proper monitoring of his body done.

There is no specific limits to the user base of the application that we propose as this can be used by anyone to diagnose their symptom. The app will take into account their personal information as well. Thus we deem it to be a universal application that will be helpful to most age groups (14 years to 65 years) and sections of

society as the benefits are common to all, that is to diagnose their disease. People of ages below 14 and over 65 may require much more specific medication as their conditions may be varying and chances of giving wrong diagnosis is high. Also our diagnosis must ensure that people with existing conditions find it suitable to use the medicines being proposed as they might hamper other medication that is already being consumed.

While developing we need to keep in mind that certain symptoms and information given by the person may be a bit exaggerated and hence may lead us to misinterpreting the symptoms. Also people with prior conditions may also forget to mention their already existing problems and allergies in which case certain medications may be unsuitable. Thus the diagnosis given must be careful and provide sufficient information for the person to understand if it is suitable for him or not.

Plan

8th February – Come up with a basic document and decide the Functional Requirements and Non-functional Requirements.

15th February – Come up with a basic design of the system and how to execute the functional requirements and an idea of the classes and objects to be used. And also where to collect data from and research about the project.

1st March – Basic understanding of GUI programming and discuss implementation of the front end and start working on it.

8th March – System Analysis and Design. Update them and decide changes to be made and implement them.

29th March – Implementation and Demo.

5th April – Testing and Demo.

12th April – Final Submission and Demo.

Functional Requirements

1) Registration:

- **Input:** A new user has to provide his/her email address, password and details regarding his/her medical history.
- **Output:** User is assigned a username
- **Pre-condition:** User should not be already registered

2) Login:

- **Input:** username and password
- **Output:** logs in the user

3) Diagnose:

- **Input:** takes the medical symptoms of user as input
- **Output:** outputs the tentative disease

4) Medication

- **Input:** Takes in the Diagnosis and Medical history.
- **Output:** Outputs generic medication for the condition and suggests whether to visit the hospital.

Non-Functional Requirements

1) Reliability: The app has to be very reliable since an incorrect diagnoses might lead to fatal consequences for the user. It must be up to date with the latest medicines, so as to give the best diagnosis and medication suggestions.

2) Security: The app has to be secure since, data leaks may end up revealing a user's sensitive personal and medical data.

3) Availability: The app has to be available to the general public, and must be advertised well.

Hardware Requirements

- Standard PC running windows/linux
- GPU

Software Requirements

- *Java*
- *Python*
- *Linux*

Publicizing the system

- Through word of mouth by letting friends and relatives know of the application.
- Put up posters to spread knowledge about the app. Make use of Fests and other events to let know students and people outside of college aware of the application.
- Set up advertisements on social media platforms and have posts boosted, liked and shared to let others know about the innovative application.
- Explain the cost effectiveness of using the application and also its accuracy in finding out diseases and giving diagnosis to the layman. This will ensure their interest in using the application as it has wide practical uses.

Cost Assessment

The app will be provided free of cost to the user, as we wish to help the general people out of good will.

Document Browser for Blind

Ankit Bagde - 17CS30009

Naimesh Pramanik - 17CS300023

Table of contents:

Introduction

- Purpose or need for the project?
- Intended Audience

Overall Description

- Existing solutions for blind/visually impaired people
- Our approach to the problem!
- What are the challenges to be overcome while developing the system?

System Features

- Functional Requirements
- Non-Functional Requirements

References

Introduction

Purpose or Need for the project?

- Blind or visually impaired people face a lot of difficulties while browsing internet.
- There is a need for such software which can be user-friendly to blind or visually impaired people so that they can browse without much difficulty.

Intended Audience

- This software is helpful for several classes of people ranging from partially blind to such people who are completely blind. Any visually impaired individual can browse internet with help of this software.

Overall Description

Existing solutions for blind/visually impaired people

Solutions to the problem of web accessibility fall into one of four categories:

- ⑩ Reliance on a conventional web browser and a screen reader
- ⑩ Utilizing the accessibility features of HTML and existing web clients
- ⑩ Using trans coding proxy servers to convert web-page HTML into a more accessible format
- ⑩ Using a dedicated web browser.

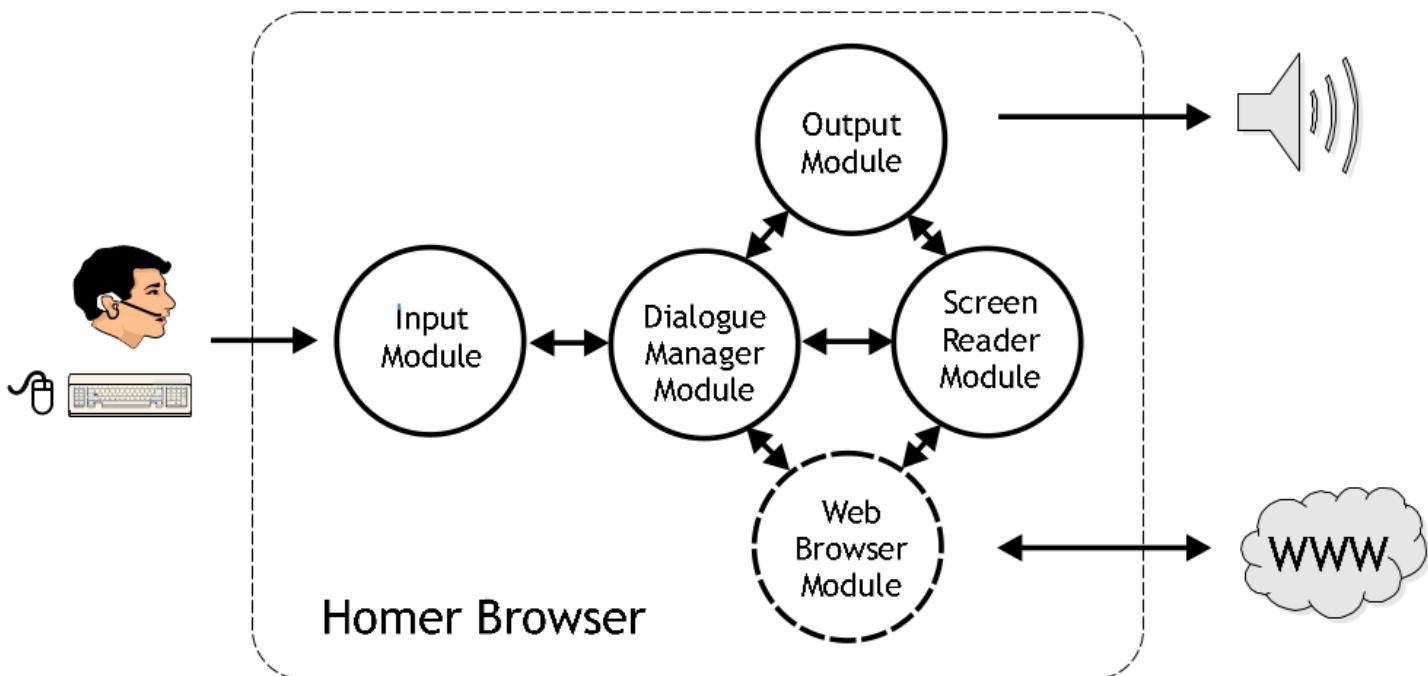
For more details on existing solutions click [here](#) and follow onwards page 4.

Our approach to the problem!

- ⑩ Our approach to the problem will be basically the development of a voice driven web browser.
- ⑩ Some shortcut keys will be allotted to users through which they can control the software for eg. next, previous, repeat etc.
- ⑩ Text to speech and Speech to text recognition techniques will be used.

What are the challenges to be overcome while developing the system?

- ⑩ If we talk regarding the screen reader method then the problem with this approach is the inaccessibility of content displayed by the browser and the complexity of the user interface. The resulting control interface can be, as noted, very complex.
- ⑩ Some screen readers use the document tree (i.e., the parsed document) as their input. However, older screen readers make use of the rendered version of a document, meaning that document order or structure may be lost (e.g., when tables are used for layout) and their output may be confusing.



Simple model of voice driven browser for blind

- ⑩ Typically our aim is to render the content of a webpage as a text-only flat document and permit the user to access this accessible content using their normal assistive technology, typically a screen reader.

System Features

Functional Requirements

- ⑩ Input from user will be taken in the form of the voice which can be converted to string. Our software can recognize certain keywords. A button(probably a switch) may be used to activate input command. For e.g.
 - Back page – Go to back page.
 - Repeat – Commanding system to read the text again.

Many such simple keywords will be used by the software to activate certain commands. Majorly, most of the [chrome shortcuts](#) may be used in order to make our software more user-friendly.
- ⑩ To read webpages correctly, web scraping technique will be used which may be useful to extract well amount of data from simple webpages. Through [JavaScript Window Location](#) we may also extract the current page address (URL).
- ⑩ After data extraction from a particular webpage we will ask user to give input over what he may like to listen about the web page.
 - Head – Software will read head tag.
 - Body – Software will read body tag.

Similarly, other specified keywords will be used to enable the reading of a particular tag.

Non - Functional Requirements

- ⑩ Performance & Usability – The software is easy to learn as it contains few obvious keywords for the user to memorize. For e.g. back page, repeat, refresh, etc. It will not take much efforts for the user to learn the software. If we talk regarding efficiency of the software as we have chosen a specific term for a particular action, the output will be accurate.
- ⑩ Adaptability & Scalability - The software is browser dependent, so it doesn't depend on the operating system used. Basically, this software will be a google chrome extension. Moreover, the software will work smoothly even if it has to handle more tabs as it has ability to access history of webpages and has ability of data backup.
- ⑩ Robustness - In case of incorrect input, the software will ask the user to speak again. In such cases, if the user falls under the category of partially blind people than he/she may use the chrome shortcuts to access the command. Being in its beta phase, the current software may not tackle complex pages. It may be used over simple head and body structure.

References

References are made from the following documents :

- www.researchgate.net/publication/221481294_A_voice-driven_web_browser_for_blind_people
- <https://pdfs.semanticscholar.org/cd19/33070b4333a28b122741a51325ff3ae5075d.pdf>

Software Requirements Specification

ExpCalc

Handwritten calculator

Version 2.0

Prepared by : Rajdeep Mondal(17CS10041)

Manas Madine(17CS10025)

**Department of Computer Science and Engineering, Indian Institute of
Technology Kharagpur**

Email: mondalrajdeep814@gmail.com

12.02.2019

REQUIREMENT ANALYSIS:

The Need:

Handwriting calculator can find a variety of uses among kids and adults alike with varying interests and skill levels. Younger elementary-age kids can write a number on the screen and then watch it magically transform into text, to build confidence with accurate number formation. Parents who might not be confident in solving difficult math calculations can use the app as a strategy to provide a solution. Children with dyscalculia, which impacts their ability to learn new math concepts, the app can be used for additional instruction and educational activities that may help improve their overall success and confidence. As with any app that readily provides answers, parents should supervise when and how it's used. Students, professors and researchers can quickly check their calculations if they use this app, as they do not always have access to a calculator.

Addressing the Need:

The plan is to use deep learning and computer vision techniques for recognizing the handwritten expressions. The basic ambition of the project is to develop a robust enough system which will take as input an arithmetic expression in the user's own handwriting traced by him on a GUI, perform the required calculations and display the output.

Prospective Users:

The prospective users of this application include kids, students, teachers, parents, professors and researchers alike. It will also include people who require it for quick, fast and small calculations so this incorporates companies, banks, shops and departmental stores.

Issues/Challenges to Overcome:

Since it involves handwriting recognition, we need to develop a neural network for recognizing the digits. But the neural network must be robust enough to be able to recognize variations in handwriting since the users include people of all ages. So, there are small kids who are just learning to write and then there are elderly people who may not have adequate strength in their hands to write with proper discernible handwriting, such as those suffering from Alzheimer's disease. So, the dataset on which the neural network must be trained must have exemplar handwriting from people of all ages. Also, deep learning networks require large datasets to train properly and produce satisfactory results. So, we would need to create a large enough dataset for the neural network to not overfit the training data and perform poorly. It will be a highly iterative process as it would require the hyperparameters to be tuned properly for it to yield the best performance. Also, training on a huge dataset would require a lot of time and higher end hardware.

FUNCTIONAL REQUIREMENTS:

1. Digital ink

When the left mouse button is pressed and the mouse dragged, the images will be drawn using digital ink. There will be a functionality as well for choosing different ink colours.

2. Neural network

The neural network will take as input a 28x28 pixel image and output a one hot vector according to the operator it has identified.

3. Multiple digit recognition

As pointed above the neural network only takes as input the image of one digit or operator. But in general, the user would be writing an expression against a white background (of the interface). So, the entire frame will be taken as input and the regions of interest will be identified by first converting the image to gray scale and then identifying the contours of each digit (or operator) i.e. the different regions of interest. Assuming the expression is written from left to right, the expression can be reformulated in memory by extracting the images one by one from left to right i.e. and processing them individually.

4. Resizing and reshaping the digit and operator images

Before feeding the extracted regions of interest into the neural network, the images must be properly reshaped by applying padding and shrinking them to 28x28 pixel size.

5. Calculator

This function will perform the required calculations and return the result.

OTHER NON-FUNCTIONAL REQUIREMENTS:

1) Human Computer Interaction (HCI)

A Graphical User Interface where the user is supposed to trace the numbers and the operators for performing the desired calculations, it will also show the calculated result of the input arithmetic expression

2) Dataset used for training the neural network

- a) For digit recognition the MNIST dataset is already available
- b) The dataset for the operators must be created.

3) Memory for storing the parameters for storing the neural network as well as to perform calculations

SOFTWARE REQUIREMENTS:

Programming Language: Python 3.6.7 / Jupyter Notebook

Libraries/Frameworks:

1. PyGame (version 1.9.4)
2. NumPy (version 1.15.4)
3. TensorFlow (version 1.5.0)
4. Keras (version 2.2.4)
5. OpenCV (version 3.4.5)

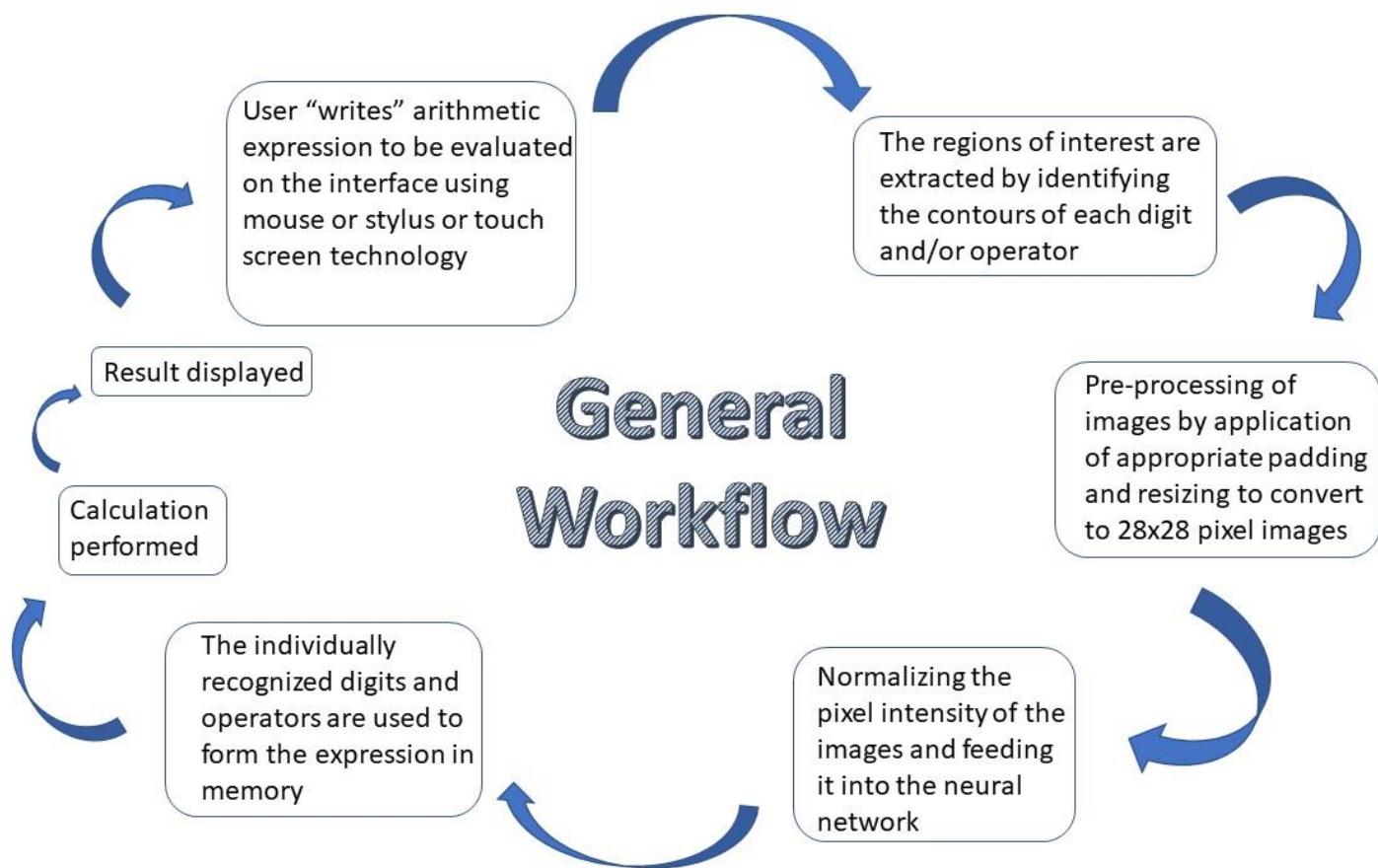
Operating Systems: Windows 10/Linux

HARDWARE REQUIREMENTS:

We will be using Google Cloud Service for storing the datasets and train the convolutional neural network using Google Colab which offers free K80 GPU, 12.76 GB RAM and 33GB storage memory.

PROJECT PLAN:

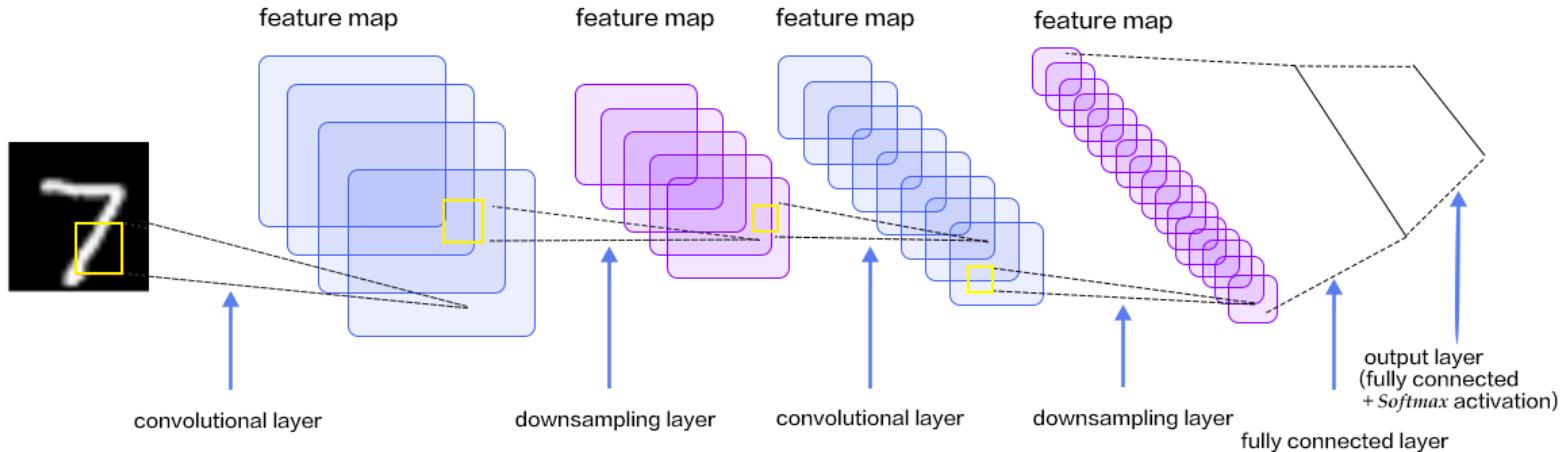
The general workflow of the software is as follows:



The project as such is implemented heavily so it is wise enough to apply a **bottom-up approach** for developing the system.

Initially, the plan is to develop the GUI using PyGame and OpenCV. The GUI will have a white background and it will use digital ink and mouse click events for tracing and drawing the digits and operators. The user can personalize the ink colour and write on the screen using his own handwriting. On the event that the left mouse button is pressed, the pointer starts drawing on the whiteboard. When the user relaxes his/her press on the mouse button, the pointer stops drawing. Post completion of writing the expression on the whiteboard, the user clicks on an ‘OK’ button at the bottom of the window.

The whiteboard frame will be converted to gray scale and the contours demarcated. The regions of interest will be extracted from the frame and resized accordingly to yield 28x28 gray scale images. All these tasks are accomplished using OpenCV.



Parallelly, the work on the machine learning part will also be going on. Roughly speaking, it will be a convolutional neural network with a terminal SoftMax layer. I since it will take time to manually select the best architecture for the model as it involves experimentation with the number of layers, tuning the hyperparameters like learning rate, batch size, optimization algorithm, etc. The neural network must be robust enough to recognize similar looking numbers, for example 3 and 8. Also, because we will be extracting the images based on their contours, so we cannot use the ‘÷’ symbol since it will be interpreted as three separate parts. So, we will be using a different symbol for this. Also ‘/’ cannot be used as it may be misinterpreted as ‘1’. The dataset of the operators will have to be generated by us as it is not available online. This dataset will be clubbed together with the MNIST dataset which is freely available online.

The expression will be read and recognized by the computer as the digit and operator images are fed into the neural network one by one. Then the expression will be evaluated, and the result displayed on the screen as a string.

Keras and NumPy libraries will be used to build the architecture for the neural network and for training and testing it. The trained model will be saved in a .h5 file whence it can be loaded and used as per requirement. The Keras library uses TensorFlow backend.

PUBLICITY AND COST:

In today's modern world what if we could simply write an equation with your hand on the graphical user interface on the screen and get the result of the arithmetic equation. Gone are those days where a user uses the stereotypical calculator on the computer in which the user uses the mouse clicking for typing even basic expressions. How cool it would be if you could simply write an equation on the screen and automatically get its output.

So here is our product EXPCALC which is a handwritten calculator in which a user simply with his hand and his own handwriting writes on our graphical user interface and gets the result of his expression and also gets digit and operation of his expression displayed.

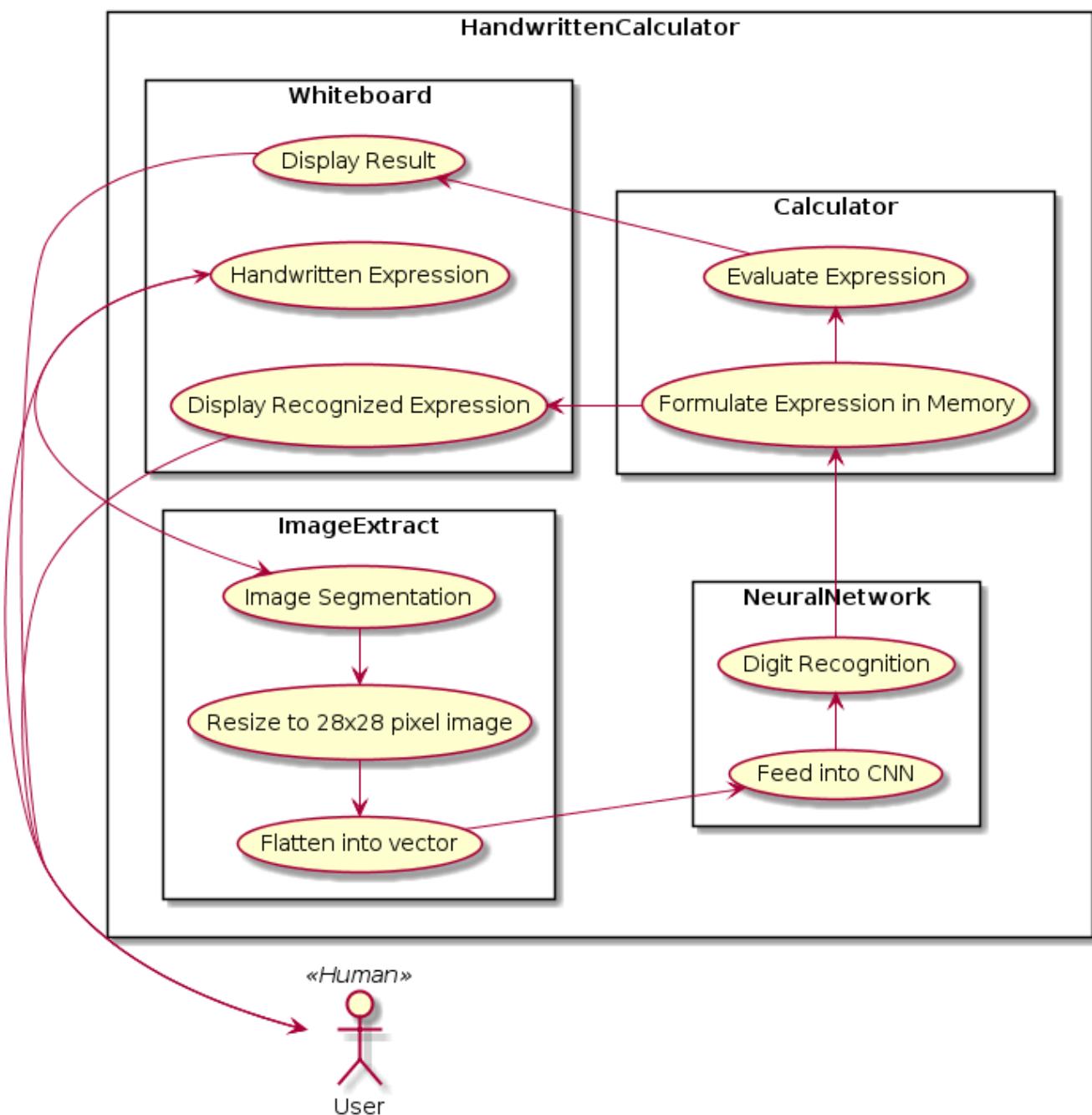
For professionals (students, engineers, professors, etc.) if there is no calculator available you can just scribble your equations on the screen and get the result of the expression, you need not worry about the neatness of the digits as our product is well trained to recognise and handle all sort of handwritings.

So our product could be used not only as a calculator but also could be used by elementary school children to learn digits as our product displays the digit written by the user. If the child writes a digit our system predicts the digit closest to the written digit if it is wrong the child learns how to write the digit properly and neatly.

So why being old fashioned and using the 90's calculators when you can use our new AI driven handwritten calculator

Cost estimate : We estimate that our product could cost around 2000 \$.

Use Case Diagram



Sharing Tour Expenses

System Requirement Specification

- Anshul Goel ,Yash Raj Gupta

About the project:

a) **Need:** On a trip of a group of people, they often have to make several payments regularly. For example, the hotel bills, the meals, the transport fares and entry tickets. Such expenses are mostly equally payable to everyone but often they are paid by individuals. Now the person has to take note of the expenses which he has made. He generally notes it down in a document or other mobile notes application. The potential problem in this system is that after the trip people can forget the expenses and they can doubt the validation of amount. Further he has to make calculation according to the expenses done by everyone else hence the calculation can become complex. After the calculation he has to ask for the money from every member and remember who has given the money. We don't want the trip to be spoiled in such calculation and unnecessary attentions so there is a need to solve the problem.

b) **Solution:** We have to make an exclusive application for this specific purpose which can perform all the needful functions in the most convenient manner so that the trip is not disturbed. The major problem with the trivial idea was to sync the data entered by the users in their respective devices. The Application would be syncing the data entered by all users and calculating the amount owed by one user to another.

c) **Prospective Users:** Users can a group of people knowing each other like a group of families or college students, employees, etc.

d) **Issues:**

- 1) Unavailability of Internet in that area.
- 2) Different OS on different devices.

Plan:

08/02/19 – 15/02/19	SRS Preparation
Mid Sem Break	
28/02/19 – 13/03/19	Learning app development and required softwares
14/03/19 – 14/04/19	Development + Modification

Functional Requirements:

- Start Trip – A user would be registering for a trip with a User name and a Trip Id.
- Add Members – The admin will add the other members in the trip.
- Add Expenses – Every member will add the expenses done by him.
- Sync Data – After every addition of expense, every member would be notified.
- End Trip – The admin will end the trip and the group would sustain until the amount is paid.
- Amount Allocation – After ending the trip the balance would be allocated to the members.
- Reminder Notification – The reminder notification would be sent to the member who have not sent the amount.

Non-functional Requirements:

- A portable device.
- Internet connectivity of the devices

Hardware/Software Used – We will be making the app using Java SQL Database Management Library.

Publicity Strategies:

- Feature App in official blog.
- Contact colleges to launch the App.
- Cost of product: \$1000

Use Case Design:

```
@startuml
```

```
top to bottom direction
```

```
skinparam packageStyle rectangle
```

```
actor Admin
```

```
actor User1
```

```
actor User2
```

```
rectangle Step1 {
```

```
User2 -> (Login)
```

```
Admin --> (Login)
```

```
User1 --> (Login)
```

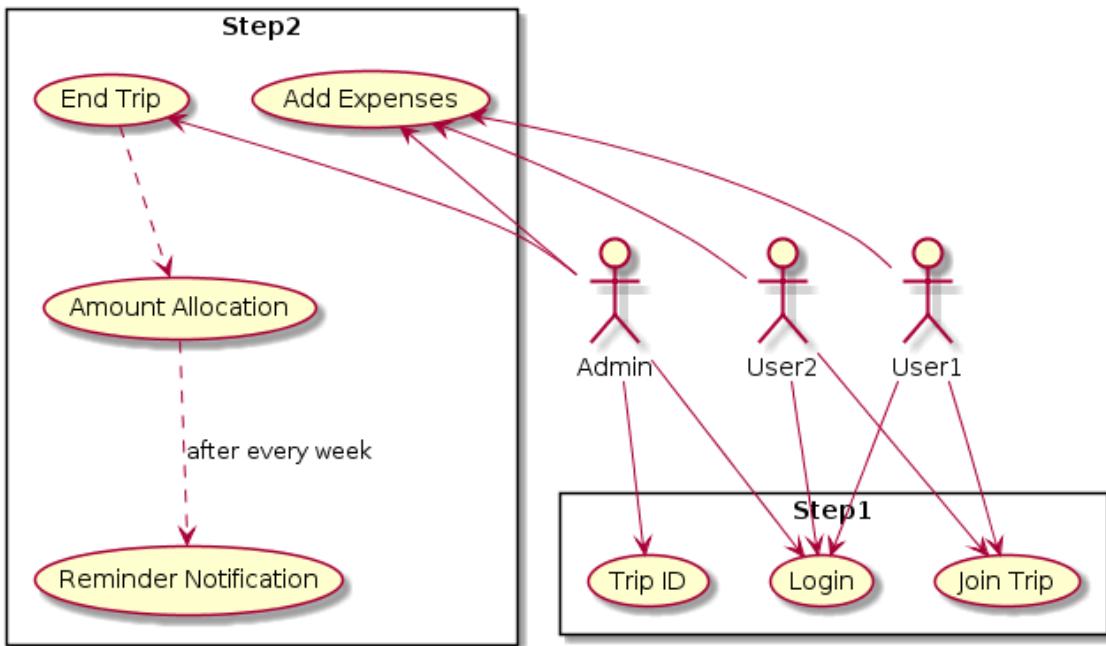
```
Admin -> (Trip ID)
```

```
User1 --> (Join Trip)
```

```
User2 --> (Join Trip)
```

}

```
rectangle Step2 {
    (Add Expenses) <--Admin
    (Add Expenses) <--User1
    (Add Expenses) <--User2
    (End Trip) <-Admin
    (End Trip) ..> (Amount Allocation)
    (Amount Allocation) ..>(Reminder Notification):after every week
}
@enduml
```



Software Requirement Specification for ONLINE C PROGRAM EVALUATION

Name of the Project:

Ccr@ckK

**BY
K VARUN RAJ
17CS10019
and
T MANITEJA
17CS10053**

1. Introduction

Online evaluation is a online system used to test correctness and efficiency of programs written in various programming languages. They are used on programming competitions held online as well as on site. Most importantly they are used to practice for a programming contests. These systems are used by lots of people, particularly students, to learn issue related to programming languages, algorithms, data-structures and to improve programming skills through training and competing in several programming contests. Furthermore, they have been used by recruiters of the well known companies for job applications.

1.1 What is the need?

We know that programming language and algorithm design are important courses in computer science which are the foundation of software development. Evaluating the program written shouldn't dependent on the availability of a specific system or software. No matter when and where students are, they should be able to focus on algorithm accuracy and running time. It is helpful for students independent learning. Competitive programming is a mind sport usually held over the internet or a local network, involving participants trying to program according to provided specifications. This project deals with the evaluation of time complexity and space complexity of a program.

1.2 Addressing the need

Programmers generally use IDE's (Integrated Development Environment) to test their programming skills. But an IDE requires a

platform to work on where as an online evaluator doesn't require any platform. It is accessible on any device from anywhere through internet. Moreover, it provides an environment for training and practicing algorithms, data-structures and programming languages.

1.3 Prospective Users

Developers : in order to be sure they are developing the right program that fulfills their requirements like time and space complexity.

Contestants, Users : to get familiar with the idea of the programming.

Judges : to get familiar with procedures involved in assigning contest problems and grading them according to their evaluation pattern.

Administrators : in order to know exactly what they have to expect from the system, right inputs and outputs and response in error situations.

1.4 Issues/Challenges to be overcome while developing the System

The main challenge to overcome would include managing the errors in the program submitted that cross the bounds. Checking for time and memory limit of the program so that it might not use up all the memory space. Other challenges include calculating the complexity and accuracy of the algorithm. We need to link an IDE to the evaluator as the interface is completely online.

2. Plan

The total work is planned to finish in a total of 6 weeks .The individual week work plan is set as follows.....

WEEK I : to abstract the software design and decide all functional and non-functional requirements that are needed to be constructed to finish the project.

WEEK II : System designing

WEEK III : Programming GUI for the software

WEEK IV : System analysis and design second stage

WEEK V : Programming for JDBC and networking

WEEK VI (+1) : Implementation , Demo and Testing

3. Requirements

There are two kinds of requirements to be discussed in this project...

3.1 Functional Requirements

The objective of this software is to simply compile , run and display the qualitative and quantitative aspects of the input code,
The following functions are to be designed ...

1. **Register()** : links to a page where the user can register himself(if not registered).
2. **Login()** : here the user can login himself to use the evaluator.
3. **Writecode()** : takes the user to a simple online text editor to type their code .
4. **Checkcode()** : this function compiles the input C code and displays the errors in it if there are any and if the code is error free , then display the complexities (space and time) of the code in a separate dialogue box .
5. **Runcode()** : this function stays frozen if there is no recently compiled code(i.e error-free) from the user. If there is a compiled code available then a dialogue box appears where the user can run his test-case inputs to get the output.

6. **CheckandRun()** : this function is basically a double step. It compiles and runs the program at the same time. The program runs only if there are no errors after compiling.
7. **Output()** : this is sub-function of both RunCode() and CheckandRun(). It gives the output of the given test case input by the user.
8. **Savefile()** : user can save his code and open it at the time of his need.
9. **Comment()** : user can post his question and ask for a solution. He can also share his views on the website.

3.2 Non-Functional Requirements

3.2.1 Design constraints :

Hardware Constraints

Database server : As the software stores the input data temporarily and user details and user files , SQL database server is required. Generally we use MySQL.

Software Constraints

Programming Languages : we are using JAVA as our constraint programming language as it is a user friendly language.

Software Attributes :

1. **Usability** : The system should be configurable from a single source at a central administrative position, and should provide a central, easy-to-use interface that will allow administrator to configure the user interface and features in a way that reduces page clutter. A system will be

considered to meet this requirement if it has a single administrative interface rather than individual links for editing each page. Furthermore, this interface must allow administrators to easily change themes and other setting that affect page layout across the entire system .

2. **Availability** :The system should be available 24 hours a day, 7 days a week
3. **Maintainability** : The system must be maintainable without substantial modification . Due to limited number of administrators and support staff, it is important that the system be mostly self-sustaining. This will reduce the number of hours spent maintaining the system and simplify maintenance tasks.
4. **Performance** : The system should support at least 100 concurrent users as the use is thought to be limited when it comes to no. of users.
5. **Security** : The security for this software is of prime importance as we store user files and details.
6. **Storage** : a maximum of 1000 users are expected to register. Each user is allotted 20Mb space which gives us an overall use of 21Gb memory space. This stores the code submitted by the user.

4. Tentative Software/Hardware environment to be used :

Software environment used are HTML (website development as the software is designed for online usage), JAVA , SQL (for data storage).

5. Advertising and Publicising the product

This product is user friendly. It can be accessible on any device. We not only provide the output of the given input but also comment on the qualitative aspects such as the time complexity and space complexity of the code. It acts as a nice platform of learning basic programming.

Project Overview

Internet based chatting has been one of the most ubiquitous technological concepts of the 21st century so far. It is one of the most important features of all social media platforms and an average human heavily invests time in it to stay connected socially with contacts in various parts of the world. Although there are a lot of chat-based platforms currently available on the internet and on the app stores of various mobile phone environments, there is still great room for improvement as the concept of chat-based platforms is still in the preliminary stages of development.

There are many desirable features which are still missing from most of the current chatting applications. An innovative chatting application could implement all of these features and greatly enhance user experience while chatting. Some of these features are real time translation to the user preferred language, self-built emoticons, recipient based predictive text, priority based notifications with an optional system overriding feature, direct web search, etc.

This application is intended to be used by the “internet dependent humans” of the 21st Century of all age groups ranging from children to elderly people. It will contribute in general to the globalization of the world by connecting humans from all social environments through a basic communication based application.

Building this application has its own set of challenges. There are network based challenges when dealing with networking applications with a large user base. It must be implemented efficiently to prevent app crashes during peak periods of activity such as during festivals. The translating feature would have to be implemented from Google Translate. There would have to be provisions against the misuse of the priority based notification feature. The personalized predictive text in a particular conversation would need a large data-set to work efficiently. Other than all these there is competition with multi-billion dollar companies such as Google and Facebook, which are also working on this.

Plan of working on the project

First of all, we intend to break the project into its fundamental parts and then we will piece them together to build the application in an object-oriented fashion. First, we will build a basic chatting environment. After that, we will allow various social media connections. Then we will implement the various add-ons such as the translation feature, the priority notification alert, the personalized predictive text, self-built emoticons, etc. Then we will upload it on Playstore so that it can be used by Android users. After that, we will market the product to widen our user base.

Functional Requirements

- There should be functions for
 - Register
 - Login
 - Logout
- This App should have friend List.
- User shall be able to clear chat history.
- User shall be able to add, block or remove friends.
- There should be an emoticon building option for the user to build their own emoticon.
- There should be a real-time text translation to other languages.
- There should be a user login interface with options to connect to other social media platforms.
- There should be a notification level setting function.

Non-Functional Requirements

- The app should be user-Friendly.
- GUI based chatting environment.
- End to end encryption for privacy.
- All data must be backed up.
- A database system to store user login credentials

Hardware Requirements

- An android smartphone running Android 5.0+
- 512 MB minimum RAM required
- Processor with speed of at least 1 GHz
- A wi-fi connection

Software requirements

- The logical aspect of this application is based on Java
- The application GUI would be done in XML
- The database storing would be done in SQL
- It will be available on Playstore in the form of an apk

Publicity

- This application can be publicized through social media platforms such as Facebook and Whatsapp
- Paid advertising on websites and other applications is also a considerable option
- We could also implement a beta version in our university to get feedback from the students on the features which could be added to improve the user experience

Cost

- This application would be available free of cost on Playstore
- There would be in-app purchases if the user wants to add some useful add-ons, which are unique to this application. This is our unique selling point.

Project code:11

DOCUMENT ARCHIVAL

by ALPESH KAUSHAL(17CS30003)
CHAITANYA BHUTADA(17CS10011)

(1)

INTRODUCTION:

This project is need to archive important document for available to view in future.

Need can be addressed by developing some way to archive documents so that user can store its document or view document anytime and saving its system space.

Prospective users can be different organization or can be done by individuals.

Purpose

Organizations archive documents for a variety reasons, in most cases relating to the overall purpose or role of the organization. Often, organizations archive documentation because it is of cultural or historical importance. Alternatively, a business may have a legal or financial reason for keeping a reliable record of past documentation, for which archiving is the most effective management method.

Challenges

Challenge is to store huge amount of archived documents and they should be secured. They should be available when the consent authority want to view archived document.

(2)

Date:	Plan:
8 February	We will prepare SRS document and do tentative planning for this Project.
15 February	We will work on design of the system.
1 March	We will work on user interface(GUI) .
8 March	We will do system analysis process and again will work on its design.
15 March	Will work with JDBC + Networking for requirement of our project , how to store archived document
29 March	Will work on its implementation (all programming work will be done) and Demo-1
5 April	Will test and Demo-2
12 April	Final Demo-3

(3)

Functional requirement:

- Login/sign up facility
- Search facility
- Save as favourite or move to top option or mark useful button

Non-Functional requirement:

- Availability: It should be available when consent authority wants to view archived document.
- Reliability:It should be safe from hackers or data manipulator.
- Recoverability : Document should be stored safely and should be recoverable when required.
- Maintainability:Database should be well maintained and particular document should be able to search when required.
- Capacity: should be able to store ,should have enough memory.

(4)

Software requirement:

- A computer running one of the following operating systems:
 - Microsoft Windows 7, 10 Windows Server 2012
- Mozilla Firefox or Google Chrome
- Java Runtime Environment (JRE) version 8 or later, Java JDBC.

Hardware requirement:

(Minimum requirements)

- 1.5 GHz Intel Pentium Processor (or equivalent).
- Minimum of 2 GB of system memory (4 GB recommended).
- Monitor : 15 VGA color.
- Ethernet connection (LAN) OR a wireless adapter (Wi-Fi).

(5)

Attractive features:

- * We will try to keep it user-friendly and try to keep it simple and easy to use.
- * We will add guide for new users to use it.
- * We will add FAQ's page.
- * Will try to make easy search results available.
- * Sorting or filtering of document based on likes or useful ways .

Software Requirements Specification

For

Garbage Collection

By

Siddana Sai Teja-17CS10051
Abhik Naskar-17CS30001

S. No	Topic	Page No
1	Introduction	2
1.1	Purpose	2
1.2	Addressing the Need	2
1.3	Prospective Users	2
1.4	Issues and Challenges	3
2	Planning and Scheduling	3
3	Specific Requirements	3
3.1	Functional Requirements	3
3.1.1	Sensors in Garbage pits	3
3.1.2	Administrator notifying the Workers	4
3.1.3	Workers informing to the Administrator	4
3.2	Non-Functional Requirements	5
3.2.1	Accessibility	5
3.2.2	Reliability	5
3.2.3	Response Time	6
3.2.4	Extensibility	6
4	Technology	6
4.1	Software Requirement	6
4.2	Hardware Requirement	6
5	Publicity	6
6	Use Case Diagram	7

1. INTRODUCTION

1.1 Purpose:

In the metropolitan cities due to insincerity of some workers the garbage deposits remain unattended leading to pollution in the residential areas. The Municipal Corporation wants to develop a software which would monitor whether the garbage deposits are filled or not and if filled would immediately attend it to clean the respective deposit.

1.2 Addressing the Need:

To address the problem certain steps can be followed:

1. A sensor can be placed at each garbage deposits which would detect whether the garbage deposits are filled or not.
2. All the sensors shall be inter-connected through a common server which would be monitored by the administrator.
3. As soon as one of the garbage deposits has been filled the sensor would send signal to the server by which the administrator would be informed about the status of the deposits.
4. An Application can be developed which would update the status of each garbage deposits that would connect the administrator with the garbage collectors so that they can take the necessary actions.
5. After the garbage has been collected by the workers they would update the status of the particular garbage deposits which would be ensured by the administrator about the completion of the work thus solving the problem.

1.3 Prospective User :

The prospective users would mainly include:

1. The Administrator
2. The Garbage collectors

1. **The Administrator:** The admin would monitor whether the workers are working sincerely and whether the garbage deposits are cleared properly and in a regular interval.
2. **The Garbage Collectors:** The garbage collectors on receiving information from the administrator would take the necessary steps and after completion of their work would inform the administrator who then would take the necessary steps.

1.4 Issues and Challenges:

We need to develop the app as a part of this software to send notifications. We should make sure that sensors and this app is working perfectly or not.

2. Planning and Scheduling of the work:

Week Number	Date	Planning
Week 1	08.02.2019	Development of SRS Document for the project
Week 2	15.02.2019	Design of System – Use Case Diagram
Week 3		
Week 4		
Week 5		
Week 6		
Week 7		
Week 8		
Week 9		

3. Specific Requirements:

3.1. Functional Requirements:

It covers the main functions that should be provided by the system. When expressed as *user* requirement, they are usually described in an abstract way.

- 1. Sensors in Garbage Deposits:** The sensors in garbage would send digital signals to the administrator once the garbage deposits are filled or emptied.

Input	Output
Weights of garbage dumped	Digital Signal 1. 1-if filled 2. 0-if not filled

2. **Administrator notifying the Workers:** The administrator on receiving information from the servers would send an SMS or some notification to the workers who on receiving the message would act accordingly.

Input	Output
Digital signal on server as send from the garbage pit	Send notification to the workers to take necessary actions.

3. **Workers informing to the administrator:** The workers on completion of work would send the information to the administrator and simultaneously the sensor attached there would send information to the server. The administrator on matching both the records would update the same on the application and thus the corresponding work would be completed.

Person	Input	Output
Workers	Notification about which garbage is filled	Attend the garbage pit to clean it and after completion send the

Person	Input	Output
		information to the administrator.
Administrator	1. Info about cleaning status from the workers. 2. Sensors giving info to the server about the status of the corresponding garbage pit.	Update the status of the garbage pit on the application.

3.2. Non-Functional Requirements:

Non-Functional Requirements describe how the system works. They serve as the constraints or restrictions on design of the system.

1. **Accessibility:** This software should be accessible only to the workers and the administrator. Updating the status of the garbage pit should be accessible only to the administrator.
2. **Reliability:** Even after using it for a sufficiently long time, it should work accurately, else the man hours and the amount of money spent will be in vain.

3. **Response Time:** Once the garbage pit is full, sensors should send digital signal to the main server as soon as possible. Also, the notifications sent by the administrators to the workers and by the workers to the administrator should reach immediately. And the status of the garbage pit should be modified as soon as the administrator changes it.
4. **Extensibility:** This software and the corresponding application should be developed by taking future growth into consideration. It should be developed in such a way that it should be easy to modify the existing functional requirement and also to add the new functional requirement if at all required.

4. Technology:

4.1 Software Requirement:

1. JAVA:

Since we are developing an application, Java language is sufficient.

4.2 Hardware Requirement:

1. PC:

At the Administrator side for the server with built in Operating System.

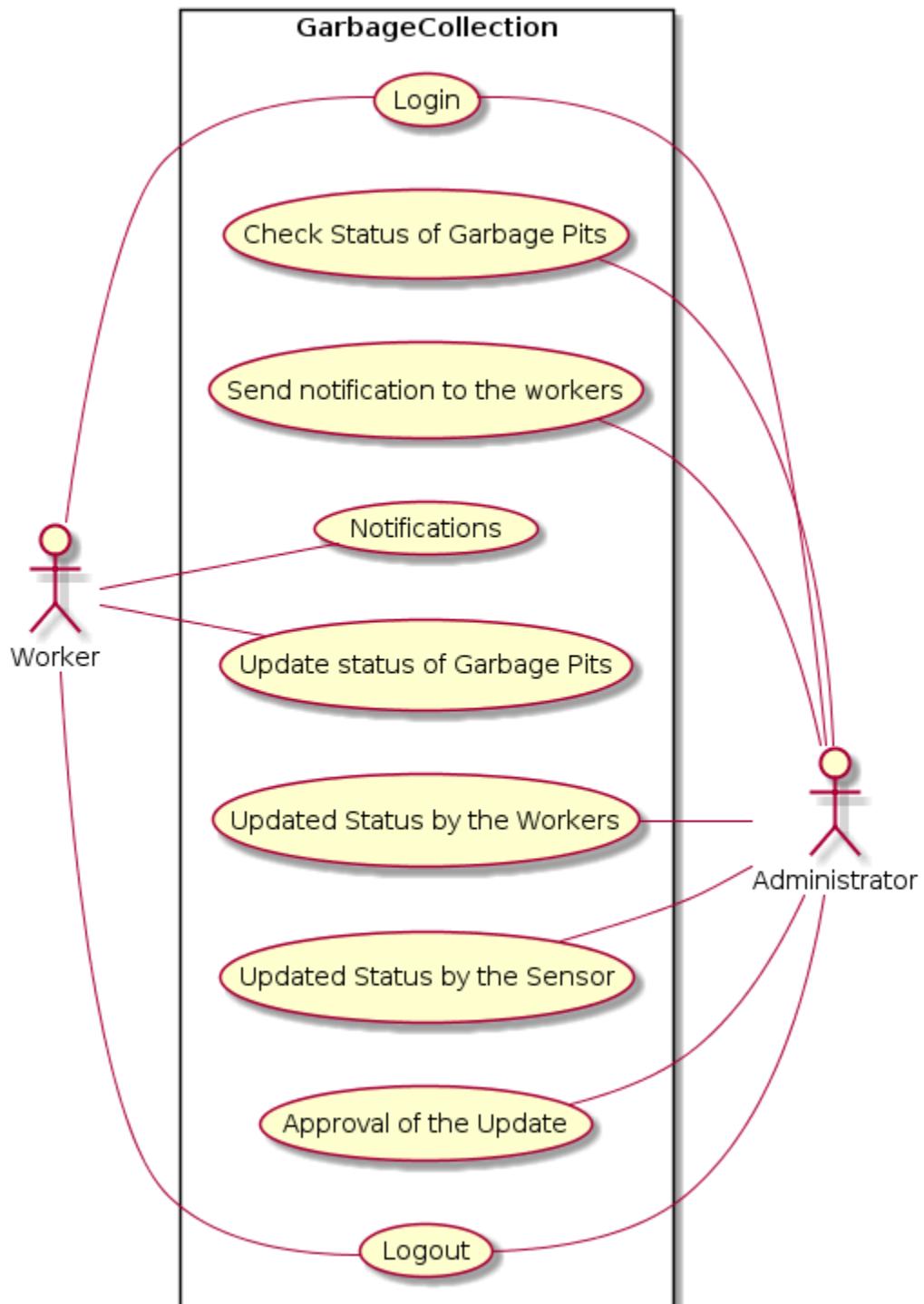
2. Smart Phone:

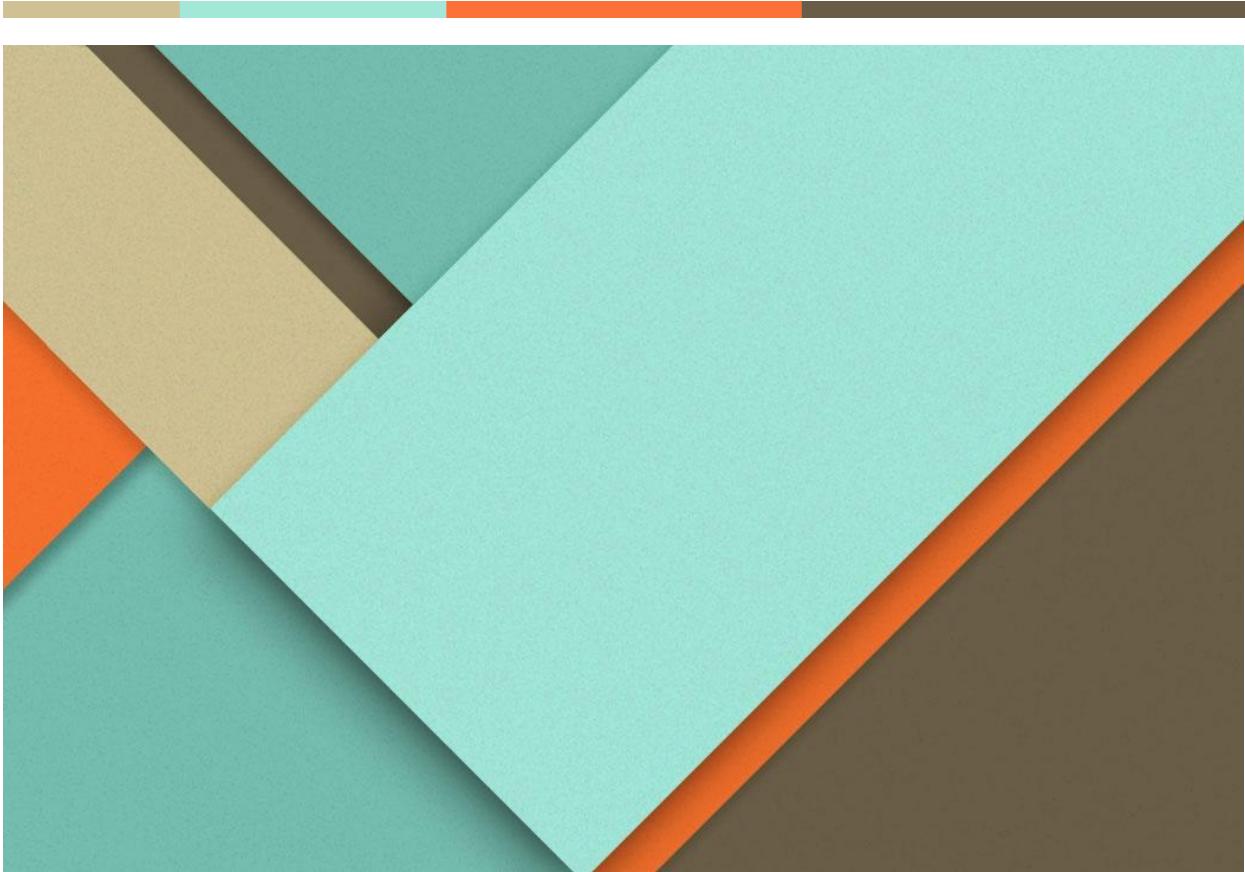
At the workers side, with at least 1 GB RAM.

5. Publicity:

Since this software is developed for the Municipal Corporation and not for the public/people, publicizing is not necessary. But for making the Government to implement it, we would explain the advantages of this software like how we can make hygienic and ecofriendly streets and colonies in an easy manner and with the less resources with the help of this software.

6. USE CASE DIAGRAM





FOOD SUPPLY(Web-Based App)

Initiated - 09.04.2019

—

Akash Tiwari || Prashant Shishodia

17CS10003 || 17CS10035

IIT Kharagpur

Overview

Web Based app for Ordering food from restaurants in vicinity with online payment options and ratings.

Need-Nowadays a major Barrier for increasing customers in restaurants is reach and distance, Often customers are in confusion over various restaurants due to food items offered, quality and prices.

Addressing needs- This app will resolve this issue by bringing all menus of the restaurants in vicinity of the user in one place, with prices and user given ratings, thus giving the restaurants more reach and the customers for options .

Prospective users- We will target mainly corporate employees, College Students and faculties. As these people often do not have access to "tasty" and healthy food, Or are short on time to search for it.

Issues/Challenges- Some of the problems faced will be, Dynamically updating attributes of a particular food item, whether it's still available or not, price changes , closing particular web-page when restaurant closes,Payment processing,Getting Delivery times according to different times.

Goals/Milestones

1. We plan to cover the project or present a working release in 7-8 weeks.
2. **Week 1-** SRS documentation, user case diagrams, identifying stakeholders,Discussion of Features to be added.
3. **Week 2-** Basic layout, adding Pseudo-lists of food items of Pseudo-restaurants for simulation. Implementing a Cart and adding feature to add food items to it.
4. **Week 3 -** implementing Navigation for navigating between different restaurant pages. Add options and features for restaurants to manage their page, like food item addition/deletion/change price/item out of stock, Delivery time based on time.

- 
5. **Week 4** - Adding payment processing and address addition, Sending Orders placed to particular restaurants contact details like email.
 6. **Week 5** - Add Review Feature and Begin work on Front-end. Designing the Web-site and Web-pages.
 7. **Week 6** - Testing and improvements, Debugging.
 8. **Week 7** - Scope for extension.
 9. **Week 8** - Deployment of release 1.

Specifications/Requirements

Functional Requirements

1. **Display page of particular Restaurant(user)-**
 - a. Input=user-based(Mouse-Click)
 - b. Output = True if successfully displayed else false
 - c. Navigates to the particular Restaurant page.
2. **User Sign-up/Sign in**
 - a. input=email-address.
 - b. Output= Session ID.
 - c. Allows the user to sign-up or Log-in.
3. **Add Food item to cart(user)-**
 - a. Input= Quantity via Keyboard
 - b. Output=Price of the items.
 - c. Takes a number as input (quantity) and outputs the total price for it.
4. **Delete Food item from cart(user)-**
 - a. Input= User's Mouse click
 - b. Output=True if successfully deleted else false.
 - c. Deleted the clicked Food item.
5. **Check out-**
 - a. input= User's Mouse click
 - b. output>Returns total Output price.

c. Function/class to evaluate final amount to be paid and send the order.

6. Display page of particular Restaurant(Restaurant manager)-

- a. Input=user-based(Mouse-Click)
- b. Output = True if successfully displayed else false
- c. Navigates to the particular Restaurant page.

7. Add New Food item(Restaurant)-

- a. Input= New name and Price of Dish.
- b. Output=True of successfully added else false.
- c. Allows restaurant to add a new dish.

8. Delete Food item(Restaurant)-

- a. Input= User's Mouse click
- b. Output=True if successfully deleted else false.
- c. Allows restaurant to delete existing dish.

9. Change Price(Restaurant)-

- a. Input = integer as new price per item.
- b. Output- True if successfully changed else false.
- c. Allows Restaurant to change price of existing dish.

10. Change Status(Restaurant)-

- a. input= New status
- b. Output= New status of item
- c. Allows restaurant to change the current status of food items,like out of stock.

11. Change Restaurant status.

- a. Input - user click.
- b. Output= True if status changed successfully else False.
- c. Allows restaurant to change their status to closed or open.

12. Obtain Help

- a. Input = User Query (Keyboard)
- b. Output = True if successfully submitted, else false
- c. Send the query to administration

13. Feedback/Rating

- a. Input = Mouse Click
- b. Output = True if rating is successfully submitted, else false
- c. Submit user rating into database

14. Modify Restaurant

- a. Input = Mouse Click
- b. Output = True if status is successfully modified, else false
- c. Update restaurant status in database

15. Approve Restaurant

- a. Input = Mouse Click
- b. Output = New Restaurant ID, if successfully approved else false
- c. Adds Restaurant into database

16. Resolve Complaints

- a. Input = String (Admin entered string)
- b. Output = void
- c. Resolves user complaints

17. Get Order

- a. Input = Employee Id
- b. Output = Order list
- c. Returns order list for particular employee

18. Update Order Status

- a. Input = Order Id
- b. Output = True if successfully updated else false
- c. Updates order status on delivery

Use-Case Diagram on next page

Non-Functional Requirements

1. Time Requirement- Backend to be completed by ~5 weeks, Frontend ~6 weeks and Deployment of release 1 by end of semester(week-7 or week 8).

- 
2. Manpower=Contributors Limited to 2 people.
 3. Resources at Disposal- Software Engineering Laboratory, Teaching Assistants, Internet.

Hardware/Software

I. Software Environments

1. Java lang. along with libraries in c,python,c++.
2. HTML & CSS for Front-end development/Web-site.
3. Sublime Text editor & Javac Compiler.
4. Github for maintaining project.

II. Hardware Requirement

1. Server(for final release) to keep the website running all times.
2. Laptops/PCs with Networking.

Publicising

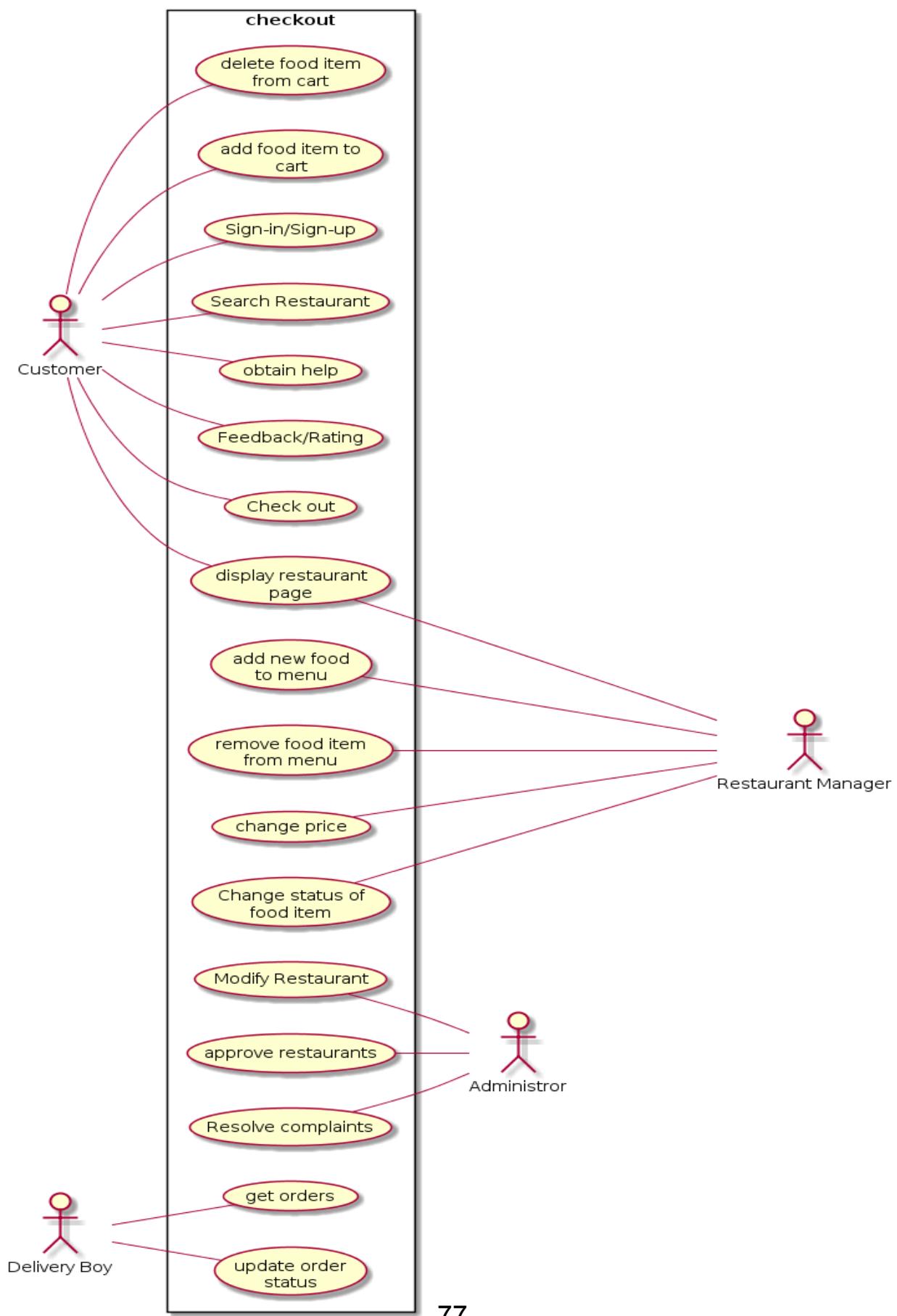
1. Dynamic update of data & availability in real time is one of the special features of our product.
2. Giving coupons who order occasionally from our website.
3. Partnering with restaurants to advertise and online partners to publicise our website.

Cost

1. In terms of Time=2-3 Months.
2. In terms of number of developers=2 developers.
3. In Terms of Money=Cost of advertising and partnering with restaurants.

Use-Case Diagram

On next page/slide





Jedi-t

1) What is the need of a multimedia editor?

The current solutions to edit documents are widely keyboard based, which -

- a) Is not intuitive to use and
- b) Poses a challenge to users who are not well-versed with using a structured keyboard (physical or virtual).
- c) Specially abled user that have hand problems who are unable to type.
- d) Editing documents also requires a specific position of the body with respect to the device
- e) Specially abled users such as blind people are unable to use physical keyboards.

Currently, it is difficult for blind people to interact with documents as they cannot read them.

2) How the need can be addressed?

A multimedia editor will enable such people to write naturally on their touch screen and it will automatically detect their handwriting and create documents.

This will eliminate the need for users to be in a specific orientation with respect to the device as well as create an intuitive medium for the user.

The proposed multimedia editor will also possess speech-to-text abilities so that blind people as well as people with hand disabilities can also create and edit documents with ease.

Text-to-speech system of the multimedia editor will allow better interaction of blind people with the document as they can have the document read to them.

3) What sets it apart from regular text editors?

Although some disable-friendly editors are available in the market, these are not tailored for general users and power users.

We aim to create a one-for-all solution with such features that makes it usable by all people alike:

- a) Ability to save multiple items in the clipboard, with shortcuts to paste them individually or all at once.
- b) User customizable keybindings
- c) Ability for automatic document uploading to the cloud and easy link sharing

4) Who will be the prospective users?

We expect the following users to be able to take advantage of a multimedia editor over and above the existing keyboard-based editors -

- a) Users who are not familiar with keyboard input system (virtual or physical)
- b) Physically disabled users who are unable to type

- c) Blind users who cannot use the keyboard and cannot see the document
 - d) People who are unable to sit at the same composure to type for long hours
 - e) Regular users who want to use keyboard-based document editing
- 5) **What are the issues/challenges to be overcome while developing the system?**

We anticipate the following challenges that we may face while developing the system -

- a) For handwriting recognition system, the multimedia editor should be able to identify a wide variety of handwritings, suggest dialect, grammar and spelling correction,
- b) For speech-to-text abilities, the system must be robust enough to understand different variations of the same words spoken in different dialects.
- c) For text-to-speech abilities, the rate and accent of the spoken voice should be modifiable so that the user can clearly understand what is being spoken.
- d) For such a complex software, we have to make sure it has a quick response time, does not demand much computational resource and can manage resources efficiently

Working stages for the development of the project -

- 1) Stage 1 - Prepare the Software requirements Specification (SRS) Document
- 2) Stage 2 - Design the process of development of the system and the involved subsystems.
- 3) Stage 3 - Implementation of text-to-speech system.
- 4) Stage 4 - Implementation of
- 5) Stage 4 - Testing
- 6) Stage 5 - Deployment
- 7) Stage 6 - Maintenance

Identifying the requirements

Functional requirements -

- 1) The system should be able to convert the standard British English Language from speech-to-text in realtime, at least with a speed of 150 words per minute and an accuracy of greater than 95% tested on standard tests.
- 2) The system should be able to convert the standard British English Language handwritten text to digital text, with a minimum speed of 1000 words per minute and an

accuracy of greater than 90% tested on standard tests.

- 3) The multimedia editor should be able to convert text-to-speech at the rate ranging from 80 words/minute to 200 words/minute in standard British English Language.
- 4) The editor should possess basic word editing abilities such as saving a document in most popular formats, renaming documents, saving and auto-saving of documents, opening documents, merging documents, changing font size, colour and font family, autocorrect and auto-suggest, print, undo and redo, cut, copy and paste.
- 5) The ability to add, resize and crop images should also be present, along with the ability to bullet points, number points and indenting lines and paragraphs.

Non-functional requirements -

- 1) A team of 4-5 people with at least 2 years of experience in software development and preferably experienced in areas pertaining to use of speech recognition, handwriting recognition, and/or speech-to-text systems.
- 2) The software is a mobile application that is available on both Android and iOS platforms.
- 3) The software should be built in a hard deadline of 30 weeks, at which time, should be ready to be launched in both platforms' respective app stores.
- 4) The total cost of development should not exceed \$10,000.

Tentative hardware/software requirements -

- 1) For the software development process, Xamarin should be used for cross platform app-development so that both Android and iOS apps can be built simultaneously.
- 2) The multimedia editor should run on all versions of iOS 10 and above.
- 3) The multimedia editor should run of all versions of Android 8 and above.

How would you like to publicise your system (i.e., pointing out the attractive features, novel things, etc.). Also, what would be the cost of product according to your assessment.

We would like to publicise the following unique and novel features to the mentioned user-base -

Feature	User-base
Handwriting recognition	Users who are not familiar with keyboard input system (virtual or physical)
Speech to text	Physically disabled users who are unable to type
Speech to text and text to speech	Blind users who cannot use the keyboard and cannot see the document
Speech to text	People who are unable to sit at the same composure to type for long hours

Power editing features	Regular users who want to use keyboard-based document editing
Automatic cloud uploading and link sharing	Users who have to regularly share their documents

Our target audience can be divided into two sections -

1. People well versed with technology - For advertising to such people, modern ad methods such as tech websites, TV ads, newspaper ads etc.
2. People who are not comfortable with technology - For advertising to such people, we have to fall back to traditional advertising methods such as radio, bus stands, parks and newspaper ads etc.

An initial guess for the total cost of development is \$8,500.

Software Requirements Specification

for
Hindi Editor

Version 1.0

Prepared by: Bhukya Rahul and Gaurav Goyal

IIT Kharagpur

Group 15

<08/02/2019>

Table of Contents

1. Introduction	1
1.1 Purpose	1
1.2 Idea	1
1.3 Prospective Users	1
1.4 Issues and Challenges	1
2. Working of Project.....	2
3. Functionality.....	2
3.1 Functional	2
3.2 Non-Functional	2
4. Project Requirements	2
4.1 Software	2
4.2 Hardware	2
5. Publicizing Project	3

1. Introduction

1.1 Purpose

- Hindi is almost the 3rd most spoken language in the world.
- In India most of the people can speak and write in Hindi but not in English.
- In India most of the computers demand English hence these leave millions of potential users in the sidelines who could not write and read English.
- Most of the text editors are in English and only English literate people can use them.
- For English illiterate people to get familiar with editors.
- With use of this editor, they can document important things in Hindi language and won't be dependent on others to do same.

1.2 Idea

We can address their need by developing a test editor in Hindi that will perform all the operations as performed by other editors.

1.3 Prospective Users

- :=> People who are English illiterate and know Hindi and can work with Computer to some extent.
- :=> People who teach online courses might use our editor to target these people(who knows Hindi but not English) for teaching them and improve their business.
- :=> People who teach online courses might use our editor to target these people(who knows Hindi but not English) for teaching them and improve their business.
- :=> Even people who know both Hindi and English might like to use Hindi editor as they might have some emotional connect their Mother language.

1.4 Issues and challenges

- The editor will be a desktop application in java.
- The application should have a Hindi keyboard as a virtual software in it.
- All the interface should be in Hindi.
- It should be able to save files as done in normal editor and should support basic functions of a normal editor.
- Displaying Hindi text on screen.

2. Working of Project

- 1) Learning the requirements of project (Desktop application in Java, Developing a virtual Hindi keyboard, Parsing Hindi in Java)
- 2) Create a virtual Hindi keyboard and check it with any available editor.
- 3) Writing code for Parsing Hindi and incorporating with virtual keyboard.
- 4) Developing a desktop application and incorporating parsing and keyboard with it.
- 5) Improvising the application by adding various features as present in normal editor.

3. Functionalist

3.1 Functional

-Taking input in Hindi :

Since the editor is a Hindi editor so we need to take input in Hindi only with help of virtual keyboard.

-Displaying input in Hindi on screen :

The editor should be able to display given input in Hindi on screen.

-Parsing Hindi queries :

Various queries like search, replace should be supported by editor.

3.2 Non-Functional

-Features of a general editor :

Able to save a file, open a new document and customized display.

4. Project Requirements

Both hardware and software requirements are there to build Hindi Editor.
Requirements are mentioned below.

4.1 Software

- 1) Java Foundation Classes :-> To develop GUI of our application (example: Swing or AWT).
- 2) Java Applet :-> To create Virtual Hindi keyboard
- 3) Inter applet communication to communicate between keyboard and editor.
- 4) Apache Commons Lang :-> To display in Hindi in editor.

4.2 Hardware

- 1) Basic system requirements.

5. Publicizing Project

- The software aims at helping people who are literate only in Hindi and want to document things using Hindi.
- It can be used as a basic editor in offices where documentation need to be done in Hindi.
- Click based keyboard to avoid complications of general Hindi physical keyboard.
- The cost of product according to me is around 1k dollars.

Software Requirements Specification

for
Bona fide

The Plagiarism Checker

Version 1.0 approved

Prepared by Ayush Kumar and Aadarsh Sahoo

Department of Computer Science & Engineering

Indian Institute of Technology Kharagpur

8th February 2019

Table of Contents

Table of Contents

Revision History

1. Introduction

- 1.1 Purpose and Need
- 1.2 Addressing the Need
- 1.3 Prospective Users or Intended Audience
- 1.4 Issues and Challenges to Overcome
- 1.5 References

2. Work Plan for the Project

- 2.1 Timeline

3. Functional Requirements

- 3.1 Application Startup
- 3.2 User Registration
- 3.3 User Login
- 3.4 Naming a Session
- 3.5 Text Input Function
- 3.6 Upload File Function
- 3.7 Extract Function
- 3.8 Search in the Search Engine Function
- 3.9 Web Page Loader and Parser
- 3.10 The Sentence Searching Algorithm
- 3.11 Sentence-wise Result Function
- 3.12 Matched Source Result Function
- 3.13 Final Result/Summary Display Function
- 3.14 Report Download Function
- 3.15 New Search Function
- 3.16 Session History Display Function
- 3.17 Logout Function

4. Nonfunctional Requirements

- 4.1 Performance Requirements
- 4.2 Security Requirements
- 4.3 Design Constraints
- 4.4 Software Quality Attributes

5. External Interface/Environment Requirements

- 5.1 Hardware Interfaces
- 5.2 Software Interfaces

6. Attractive System Features

- 6.1 We do not Save your Precious Data
- 6.2 We provide a FREE Trial with NO feature restriction!
- 6.3 Easy to Use
- 6.4 Personalised Sessions with User Accounts
- 6.5 Save your Session History

6.6 You may contact us for Advertisements

7. Estimated Cost for the Product

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose and Need

Plagiarism is the "wrongful appropriation" and "stealing and publication" of another author's "language, thoughts, ideas, or expressions" and the representation of them as one's own original work. Plagiarism is a form of academic dishonesty or we can say that it's "Cheating". With the rise and spread of the Internet and the development of advanced search engines, Plagiarism has become a very serious issue in today's educational system and professional environment.

A lot of Surveys were conducted all around the world and it was found that a considerable percentage of High School and College Students admitted having cheated on their work one or more times in some form. Detailed Statistics of the surveys could be found at <https://www.plagiarism.org/article/plagiarism-facts-and-stats>.

The issues mentioned above calls for a solution. The detection of Plagiarism is near to impossible for a human being because of the enormous amount of data and resources available all around, this gave birth to the class of software known as Plagiarism Detecting Softwares. The main goal of these systems is to promote and sustain the 'value' of 'Intellectual Property' and 'Originality in Idea and Expression'.

1.2 Addressing the Need

The need can be addressed in several ways. The solution we are using in developing our software is by using the internet to search for the 'matches' between the sentences present in the document or text with the content available on the internet. This can be achieved by extracting the input text from the document and dividing them into groups/sentences, which could be searched in an advanced search engine like Google Search. Google Search with its extremely good PageRank Algorithm shows the possible websites with the searched content in an appropriate order. Now each website can be visited in order, for comparing the searched content and detecting any plagiarism involved. The output could be presented to the user with the addresses of the possible sources from where the information may be copied. The percentage of plagiarism present in a document could also be calculated and presented to the user. A 'threshold plagiarism percentage value' must be given by the user so as to classify the analyzed document as 'Plagiarised' or not.

1.3 Prospective Users or Intended Audience

This application software mainly targets Academia for its prospective users, who may use this to detect academic dishonesty and copying. The application can be used by anyone(e.g. a verification officer) who is in a position to evaluate or screen any kind of documents or publications, to determine the 'authenticity' of the submitted document as per the guidelines for the user.

1.4 Issues and Challenges to overcome

The issues and challenges that are going to arise while the development process of this software can be identified if we properly analyze the very basic principle which the software uses to detect plagiarism. Plagiarism detectors don't actually detect plagiarism, what they

actually do is detect sections of identical texts i.e. it can only detect copying or similar phrases. So the following would arise as the prospective problems:

- **Synonym Matching:** The person may copy from a source and change some of the words to their corresponding synonyms to get themselves screened from the application software.
- **Non-Verbatim Plagiarism:** The user may rewrite, translate or otherwise redraft the content from the source and deceive the software system. This problem arises because Plagiarism detectors analyze the words, they don't analyze the content i.e. it can't see if you copied the idea or information even if you didn't copy the words. So this may let go of some serious plagiarised content undetected.
- **Common Phrasing:** The document or content being analyzed is very likely to contain many common phrases in the English language, which may be reported by the software system as a match even though that might be just a coincidence. So we need to implement an Intelligent comparison system which is able to overcome the problem of common phrasing.

1.5 References

<https://www.plagiarism.org/article/what-is-plagiarism>
<https://www.plagiarism.org/article/plagiarism-facts-and-stats>
<https://en.wikipedia.org/wiki/Plagiarism>

2. Work Plan for the Project

2.1 Purpose and Need

Week	Plan
Week-1	Preparation of SRS, RAS and FS related documents.
Week-2	Designing the Structure to write the program for the application software.
Week-3	Designing the Graphical User Interface for the application software.
Week-4	Designing the Algorithm.
Week-5	Designing the complete Backend of the application software.
Week-6	Testing of the Application and Error Correction.
Week-7	Refining the GUI and making it look better and more user friendly.
Week-8	Final testing

3. Functional Requirements

3.1 Application Startup

When the user opens the application the logo of the application must be displayed in the screen for 3 seconds with a welcome message and then proceed to the user login/sign-up window.

Input : When application icon is clicked.

Output : Application Logo on the Screen with a Welcome Message and then proceed to login/sign-up window.

3.2 User Registration

An User must be able to register for an account for the application after the user has provided the required credentials like email-id, name, password, mobile no., and IP Address(determined automatically).

Input : Details and Credentials of the user.

Output : A message notifying either successful or unsuccessful sign-up with reasons.

3.2.1 Email Verification

If the email address used in the sign-up process already exists with another user, there should be a prompt for the user to notify him/her about that.

Input : Email address provided during registration.

Output : A message stating 'email address already exists!' if there is an account with the same email Id.

3.2.2 Image Upload

The user is asked to upload an image for the user profile. This step may be skipped by the user.

Input : Success message from Email Verification process.

Output : Ask to upload an Image if the user chooses to do so, else continue to the main window.

3.3 User Login

Allows the user to login to the user account whenever the user enters the username(email-id) and password and allows the user to access the application software.

Input : Email address and password by the user.

Output : Successful login on email-password matching, else an Error Message.

3.3.1 Email Not Found

If email address entered by the user is not found on the server for login then there is a prompt to notify the user that the email address was not found and the user is asked to either sign-up or re-enter the email address.

Input : Email address from login window.

Output : Message of 'Email not found!...Try Again' or redirect to the Registration Window option is shown.

3.3.2 Email-Password Mismatch

If the password entered by the user while login is not correct, the user is prompted about email-password mismatch and is asked to enter the password again.

Input : Email address and password from login window.

Output : A message of 'Email-password mismatch...Enter password again!' if the password doesn't match with the email address entered.

3.4 Naming a Session

The user is asked to name the session once it is logged in as it will be saved in the session history feature.

Input : Title of the Session by the user.

Output : Stores the title of the search result and the date for future reference with no immediate message to the user.

3.5 Text Input Function

User is allowed to type any content which he/she would like to check for plagiarism and the entered text is sent to the temporary storage. This also shows the word count of the content entered.

Input : Content typed by the user using the keyboard to be checked for plagiarism

Output : Sends the text to the temporary memory and the Word Count of the content is displayed.

3.5.1 Word Limit Exceeded

If the word count of the entered text or the text in the document file exceeds 1000 then the user is prompted to reduce the word count to 1000 or less and then enter/upload again.

Note: This function has been kept with view to the speed of the application to show the result.

Input : Word Count from the entered text or from the document uploaded.

Output : Error message of word limit exceeded and asks the user to enter or upload the content again.

3.6 Upload File Function

Uploads the file from the user's computer when chosen by the user.

Input : Click on upload file button

Output : Ask the user to browse and select and upload the file to the temporary storage.

3.6.1 File Format Mismatch

If the file uploaded is in a format other than ‘.docx’ or ‘.txt’, user is prompted about it and asked to upload the file in the compatible format only.

Input : File format of the document uploaded in the upload file function.

Output : Error message if file format does not match the compatible one.

3.6.2 Word Limit Exceeded

If the word count of the entered text or the text in the document file exceeds 1000 then the user is prompted to reduce the word count to 1000 or less and then enter/upload again.

Note: This function has been kept with view to the speed of the application to show the result.

Input : Word Count from the entered text or from the document uploaded.

Output : Error message of word limit exceeded and asks the user to enter or upload the content again.

3.7 Extract Function

After the user has entered or uploaded the content this function should separate/divide the content into divisions/sentences to make them ready to be searched in the search engine for matching and saves it in a separate file.

Input : Content Entered/Uploaded by the user.

Output : Sentences extracted and saves them in a separate file.

3.8 Search in the Search Engine Function

This function one-by-one takes the extracted sentences and searches them in the Google Search and loads the search engine result.

Input : Extracted Sentences.

Output : Search Engine Results.

3.9 Web Page Loader and Parser

This function one-by-one opens the web pages loaded by the search engine and each page is parsed and the text present in it is extracted from its HTML code and the text is Isolated.

Note: This function only checks the first 5 search results only for speed and time factors.

Input : Search result from the search engine.

Output : Extracted text from each web page.

3.10 The Sentence Searching Algorithm

This function searches the extracted text from the user content in the extracted text from the web page loader and parser. If any match is found then the address of the web page and the corresponding sentence is saved and noted in a separate file, else it just repeats the process for the rest of the sentences.

Input : Extracted text from the user as well as from the web page loader and parser

Output : Saves the web page address and the corresponding statement if a match is found in a separate file, else continues with the rest of the sentences.

3.11 Sentence-wise Result Function

It goes through the file created by the sentence searching algorithm, compares with the file created by the extract function and displays all the sentences along side with the status that whether it is a copied one or unique in the output box.

Input : The file created by the sentence search algorithm.

Output : Displays sentence wise result in the output box.

3.12 Matched Sources Display Function

User is shown the possible sources from where the content might be copied along with its website address.

Input : The file created by the sentence search algorithm.

Output : A list of all the web page addresses present in the file.

3.13 Final Result/Summary Display Function

This function calculates the percentage of plagiarism and uniqueness of the document/content of the user from the file created by the sentence searching algorithm and the file created by the extract function.

Input : The files created by the sentence searching algorithm and the extract function.

Output : Summary of the user content: percentage of plagiarism and uniqueness present.

3.14 Report Download Function

The user can download an analysis report of the content/document he/she checked for using the software. The document will contain the sentence wise result, matched source websites, percentage of plagiarism along with the title of the session in pdf format.

Input : Click on the Download Report button.

Output : A report having the details of the content searched for plagiarism like sentence wise result and matched sources to be downloaded in pdf format with the name of the file same as that of the session.

3.15 New Search Function

This option makes the application ready to conduct a search once again.

Input : Click on the New Search Button.

Output : Deletes all the temporary files created in the memory and takes the user to a Fresh Input Window with a new Session Name Prompt.

3.16 Session History Display Function

Allows the user to see all the session details which were conducted by him/her till date.

Input : Click on the See Session History Button.

Output : Session History of the user with the Session Names and the Summary of each session.

3.17 Logout Function

The logout button which upon clicked asks the user if he/she wants to logout of the session. Upon approving yes, the user is logged out of the session with all the user data deleted except the session history and the window for login or sign-up appears.

Input : Click on logout button.

Output : Log-out from the software session upon 'yes' approval by the user, along with saving the session history and deleting the rest of the data.

4. Non-Functional Requirements

4.1 Performance Requirements

4.1.1 Response Time

The software should return the desired output after analysis in a reasonable amount of time, we keep that time as a maximum of 20 seconds.

4.1.2 System Dependability

It determines the fault tolerance of the system. If the system loses the connection to the Internet or the system gets some strange or invalid input or the system faces any random failure, then the user must be informed about it.

4.1.3 Prominent Results

The result displayed must be prominent and clearly specify the amount of plagiarism and the amount of unique content found in the input document.

4.2 Security Requirements

4.2.1 Secure Search

The software should search the internet for the content securely giving utmost priority to the privacy of the user's data.

4.2.2 Temporary Storage of the Extracted Text

The extracted text from the input document must be stored temporarily and securely and should be permanently deleted after the plagiarism detection session has been over.

4.2.3 User Profile should be Unique

Any user account registered for the software should be unique and no fake accounts should be present. Only one user account per IP Address should be allowed.

4.2.4 Secure Login

The system should be secure from malicious or forced login to access the software.

4.3 Design Constraints

4.3.1 Hard Drive Space

The software should not take more than 20MB of Hard Drive Space.

4.4 Software Quality Attributes

4.4.1 Reliability

The system should be reliable i.e. it should give right and accurate results for each session held by the user.

4.4.2 Internet Connectivity

The application must be connected to Internet Connection for performing the Login activities and Sentence Search for the analysis of the document.

4.4.3 Maintainability

The application should be easy to extend. The code should be written in such a way that it favors the implementation of new functions.

4.4.4 Portability

The application should be portable to different platforms i.e. it should be adaptable in different platforms.

5. External Interface/Environment Requirements

5.1 Hardware Interfaces

This application software does not have any designated hardware so no direct hardware interfaces are required.

5.2 Software Interfaces

5.2.1 Java

Java has been chosen as the programming language for the development of this application software. Java is a programming language and computing platform first released by Sun Microsystems in 1995. It is the underlying technology that powers state-of-the-art programs including utilities, games, and business applications. Java runs on more than 850 million personal computers worldwide, and on billions of devices worldwide, including mobile and TV

devices (Oracle Technology Network, 2010), so this application is intended to run on all platforms that support JAVA without the need of recompilation as JAVA is intended to let developers **write once, run everywhere**. The version which has been chosen for this project is JRE System 1.7. The version used contains important enhancements to improve performance, stability and security of the Java applications. Java is known for his large number of libraries. Indeed, Sun provides a large number of frameworks and API in order to allow a lot of diversified uses. This is why Java was probably the best choice, at least the most suitable language, for the implementation of this project.

5.2.2 Java Libraries

Many Java Libraries have been used for execution of different and a variety of functional requirements for this application software.

5.2.3 Google Search

Google Search (Google Web Search) is one of the advanced and fastest search engines presently, which is owned by **Google Inc**. Google handles around 3.5 billion search queries of different types per day. It scans websites having a particular keyword to be searched and indexes the website as per the queries to the website. The PageRank Algorithm used by Google Search performs exceptionally well which would help us get our desired results very fast. We will be using Google Search for searching the Extracted Text in the Internet and to look for the possible webpages where matches in the contents may be found.

6. Attractive System Features

This section notes down all the attractive system features we provide for our system software which would allow the user to give our software the highest preference among all our competitors.

6.1 We do not Save your Precious Data

We respect user data. Our software is designed in such a way that after analyzing the input text or document from the user, it permanently deletes the extracted text from all possible locations which were used for performing the search on the Internet.

6.2 We provide a FREE Trial with NO feature restriction!

We provide a FREE Trial of 3 Plagiarism detection sessions to every registered user. This allows the user to experience the Very Accurate and Error Free Plagiarism Detection Sessions with all the features made available by our Application Software for FREE! The user can continue with the experience by subscribing for our premium membership for unlimited quality sessions.

6.3 Easy to Use

We have designed the application software interface in such a way that it is convenient to use by any user. It is very simple and robust at the same time. The simple User Interface allows the user to comfortably use the software without going through any user manual. The function of each option displayed in the interface is also mentioned aside concisely for better user experience.

6.4 Personalised Sessions with User Accounts

We enable access to our application software only to a registered user. The user can register and create an account for him/her. This allows the user to experience personalized sessions as per the need of the user and also helps in secure text extraction and search.

6.5 Save your Session History

We allow you to save your session history with only two features i.e. the name of the session provided by the user and the plagiarism percentage result of the input document, Nothing Else (No Other Data for Security Purposes), which can be very helpful for the user to refer in the future.

6.6 You may contact us for Advertisements

We allow the users to contact us if they wish to use our platform for their Advertisements.

7. Estimated Cost for the Product

Almost all the Software Libraries and Resources we are going to use for the development of this application software are open-source so almost all of them are available for use free of cost. But it takes time to build such software and obviously, time is precious and in future we may need some funding for the maintenance of the application software. We will be charging a minimal amount of INR 2000 as a membership fee for each registered user, provided the user wants to subscribe for the premium version of the application in which the user is allowed to have any number of free Plagiarism Detection Sessions. So as an overall view, this software is Free. Our Goal is to provide the user with good and quality software which the user uses for the betterment of the society.

CLOCKED PERSONAL DIGITAL ASSISTANT (PDA)

1. INTRODUCTION

1.a What is the need?

To take notes, store phone numbers, manage to-do lists, keep track of your calendar ,perform simple or complex computations ,keep track of and accomplish things that are needed to be done.

1.b How the need can be addressed?

The user inputs information using the touch screen and he is reminded about the event and also the information related to that event at the planned time.

1.c Who will be the prospective users?

There are two general types of PDA users. The first type wants a device to keep track of PIM data, such as names, addresses, and phone numbers, as a compact digital replacement for the pocket notepad. The second type is the power user (or enterprise environment user), who is interested not only in basic PIM (if at all), but also organization of his tasks.

1.d What are the issues/challenges to be overcome while developing the system.

2. PROJECT PLAN

Week Number	Date	Planning
Week 1	08.02.19	Devolpement of srs document for the project
Week 2		
Week 3		
Week 4		
Week 5		
Week 6		
Week 8		
Week 9		

3. REQUIREMENTS

3.a Functional Requirements

- a) The application has to keep track of the time.
- b) It has to remind the user at the specified time.
- c) It should store any information input by the user.

3.b Non Functional Requirements

- a) The application should be reliable and maintainable.
- c) It should be portable and easy to use.
- d) It should be accurate.
- e) It should run on Android device.
- f) We need to keep updating the software.
- g) There has to be a bug tracker available where users can report any bugs they have encountered so that the developers can fix it in the next release.

4.TENTATIVE HARDWARE/SOFTWARE REQUIREMENTS:

- a) The app runs on the Android Platform.
- b) The core code is written in Java and extended to android using the Android SDK.
- c) The app runs on any android Device supported by the Android SDK.
- d) The Core Java is written using JDK8.

DATA ENTRY SYSTEM

About The Project

Data entry is undoubtedly one of the most important factors in different business verticals for increasing productivity & mitigating repetitive business task. There are thousands of mistake that can be overcome with simple data entry task since the program allows the user to see in real-time the data entry process as well as the process of data retrieval.

Data entry helps in reducing the stress and also helps in transfer of huge amounts of data in very less time; hence saving time and eventually money. Apart from that, another benefit includes - increase in productivity, accuracy, maximum use of most technology, etc.

We are going to build a data entry system which would cater a large group of users [mainly small scale businesses] keeping in mind of the following things:

- Create a simple UI which a normal PC user can interact with and add, edit and remove data as well as tables.
- Show data visualizations as well as present data in intuitive manner for users.
- Allow multiple accounts on a single instance of the software for different users but a single account for data addition.
- Modification of data dynamically as well as entry of data through structured files.
- Prevention of faulty entries using RegEx and other methods which would act as a second level check on the data entry.

We simply and at the same time lower the chances of false entries making their way into the database, the system will be more successful.

We will be targeting small companies as well as home users who need to maintain data for various small purposes. Since we will be providing different accounts, a single family PC can have for all. For example, the bread-earner of the family can have a budget table and he can add expenses of everyone and everyone else can view them in various manners (tabular, pictorial etc). We will be using encryption

techniques to maintain privacy and protect data tampering by different users of the system.

There are a lot of issues but the major ones are:

- Making the UI Intuitive so that even new people to computer do not find it difficult to operate.
 - Making complex functionalities simpler by abstracting the overall terminologies.
 - Data protection and reduction of redundancy of data is important in tabular method.
 - Making it platform independent for users to be able to use on any System and device.
 - Showing large data in a visually simple manner for the users to understand easily and comprehend better.
 - Dynamic creation of table columns and dynamic entry to facilitate easier and quicker data entry.
-

Tentative Timeline

We will be doing the project in five phases, and each phase will focus on a particular aspect of the program. Below is the detailed information about each of the phase:

Phase I [Research]

First phase will focus on research and learning about the tools that will be required to build the project. This will involve learning about the technologies, testing various options for the same tasks [e.g. different database technologies].

Phase II [Learning]

Once the technologies are finalized, we will be learning the basics of them. Since technologies have a lot more to offer than we will need, we will focus on getting familiar with them and then getting to know just the usage we will be needing. An example of this would be not getting into knowing “joins” in a RDBMS.

Phase III [Backend Development]

Since the project will have two areas where we will require coding; Backend to handle operations and then the User Interface with which the user will interact. We will be building

on the backend in the third phase along with taking crucial functional decisions for the User Interface. This phase will not involve any UX design decisions, and will just focus on functional design and development.

Phase IV [UI Development]

Once the project backend is built, we will move onto deciding the design and aesthetics of the user interface along with making decisions to ease the user in interacting with the functions provided. In this phase, we will also be developing the user interface and all the graphic components required by it. We will also be linking backend and UI in this phase.

Phase V [Pre-Production Polishing]

This phase will wrap up the development of the project, and will involve rigorous testing of the same. We will also be taking reviews from people and changing the UI to make it more simpler for use. In this phase, we will also be testing the application on various operating systems and developing for them. Any bugs encountered will be fixed in the same.

Requirements

Functional

We require the following functional capabilities from the project:

- Ability to interact, create, modify and delete databases
- Allow creation of a single admin account per system.
- Allow creation of User accounts by the admin.
- Validation of data entries.
- Encryption of data as well as user passwords.
- Master backup of all data.
- Retrieval and pictorial representation of data.

Non-Functional

The following non-functional requirements are there:

- Quick retrieval of data
- Intuitive User Interface
- Minimal lag in database operations
- Visual scalability (how much data can be shown in an intuitive manner)

- Keeping track of space available and giving warning accordingly.
-

Tentative Dependencies

Due to interaction with database and user, the project will have some basic dependencies that will help the Java program interact with the database as well as ease the making of a Graphical User Interface. Below are the two major Java library requirements in the project.

Java

Java is one of the best programming languages and we will be using the same for most of the code that will be written in the project. We will be using its Object Oriented nature to foster the development pace.

Database Management System

We will need a database system (such as MySQL, SQLite, MongoDB etc.) to keep a track of the users as well as the data entry by each user in the system. In this way, we can keep a permanent storage of the current configuration.

We will be using Java Database Connectivity Handler to initiate connection with the database and perform all the operations such as insertion, modification and deletion. This will also help us in creating different users and keeping their tables separate from each other. We will be creating a single database for a single user and then he can add multiple tables inside it.

Swing

Java Swing is used to create window-based applications. It is built on the top of AWT (Abstract Windowing Toolkit) API and entirely written in java. We will be using the same to build our GUI for the user and link the backend operations with the events of the interface.

Marketing

Every entrepreneur is focus on the core abilities than sparing the time on simple & redundant tasks. Large/medium scale organizations prefer to outsource its data entry tasks to reduce the overhead cost of staff, payroll, time and data management, so that the organization/company can focus on growing their business.

Data entry is also very essential for managing huge business sectors or organization to minimize your administrative burdens, save on costs and use the time of your resources for core business activities. Apart from that, another benefit includes - increase in productivity, accuracy, maximum use of most technology, etc.

The cost of product would be minimal but the highest costs would incur in maintenance and support to the users of the product.

Software Requirements Specification

for

OBCAS

(Online BC ROY Appointment System)

Present Version 2.0

Prepared by :

P. Amshmaan Varma (17CS30025)

sashank bonda (17CS30031)

Software Engineering Lab (Section 2)

Created on : 8th February

INDEX

1)	Introduction -----	3
2)	General Description -----	5

1) Introduction

1.1) General Overview

Generally hospital is filled with all people working in kharagpur.If student had to go for some ailment he has to wait in long queues for hours and also if student has gone to hospital by rescheduling all the events for the day for some specialist doctor who comes on special days, very often the doctor will be absent or there will be huge rush of people.So this software intends to provide an online platform for booking doctor appointment at BC Roy Hospital. It basically specifies the availability dates and timings of the doctor to the patients and the patient list to the corresponding doctor respectively so if doctor is going to be absent on that particular day He or She will be informed in advance by mail and the student can also go at particular appointment slot he has booked which saves lot of time for him.

1.2) Need Of the Software

Booking an appointment online can have wide uses. The patient uses the system to find a doctor and view their schedule. The patient then uses the system to request an appointment at an available time. This can reduce waiting times at the hospital and huge queues at the hospital. Doctors can also keep track of number of their patients using this software.

1.3) Addressing the needs

Patients can create the respective accounts in the website. On successful registration, they can make an appointment in the available timings. The account also keeps track of previous appointments for returning patients. Then the schedule is made accordingly in the doctors account. Patients are also notified if any emergency unavailability of the doctor as a notification in their account

1.4) Scope of the Product

This product is mainly based on the features and specifications of BC Roy hospital but can be extended to any other hospitals with some changes based on the requirements. This product can be used by students and faculty of IIT Kharagpur. Doctors and Registration Staff of BC Roy can also use this to keep track of patients easily.

1.5) Challenges while developing

The data has to be constantly updated so that the appointments don't collapse. Also should take care if the doctor has changed his timings temporarily, patients should be updated of the information.

1.6) Document Conventions

1.7) References

FUNCTIONAL REQUIREMENTS

- User must sign up with his id which is his medical book number and provide his basic details and some medical history

- User after logging into his account should be able to book appointment of corresponding doctor in particular time slot
- User must be able access the updated information of schedules of doctors ,the updation of data takes place for every 15 min
- User will be informed if his appointment is cancelled due to no time slot available or if corresponding doctor is unavailable by message in his account and also through mail
- The website also stores medical history of student during his stay in IIT kharagpur
- User can have a digital medical book

NON FUNCTIONAL REQUIREMENTS

-SYSTEM will respond in 1 sec after checking all doctors information

-The system can support 500 users at a time

-Users login id and password will be protected

-Any modification (insert,delete) like if a patient has checked out room becomes vacant so that information updation can be done only by administrator of that ward

-The System will be available 24/7

- The website will have virus protection,password management,regular database archiving

PLATFORM

It will be launched as web-based application.

SOFTWARE REQUIREMENT

front end:- html,css

Database:-MySQL Server

GUI and database java programming

HARDWARE REQUIREMENT

RAM:-1GB

HARD-DISK:-20GB

METHODOLOGY

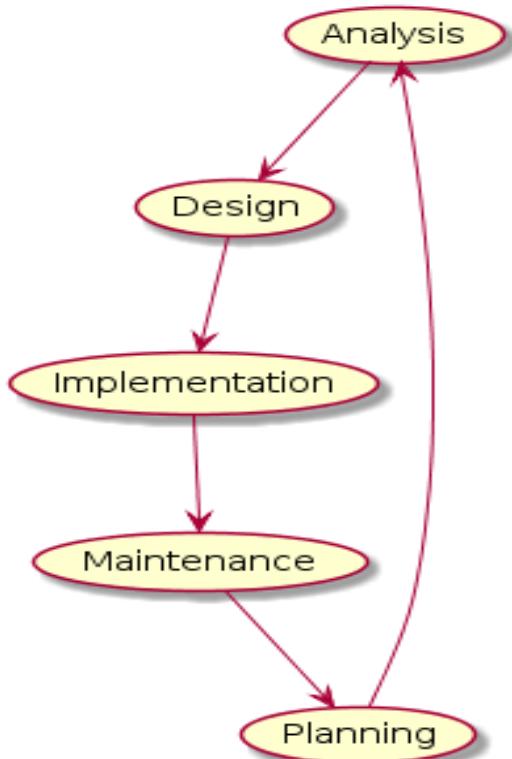
OBJECT ORIENTED PROGRAMMING

UML(UNIFIED MODELLING LANGUAGE)

USERS

- PATIENTS HAVING ERP ACCOUNT i.e STUDENTS AND STAFF etc.
- BC ROY AUTHORITY WHICH WILL ACT AS ADMINISTRATOR
- DOCTORS AND HOSPITAL STAFF
-

DEVELOPMENT PROCESS



THE PROJECT IS AIMED TO BE COMPLETED IN 6 WEEKS TIME STEP BY STEP IN 4 STAGES

-REQUIREMENT SPECIFICATION

-DESIGN

-CODING

-TESTING

RIGHT NOW THIS IS VERSION 2.0 OF SRS DOCUMENT

SYSTEM FEATURES:-

PATIENT FEATURES

To register new account and login and logout from that account. Allowed to view doctors list based on departments and their specifications. Users are to book , update and cancel an appointment. On successful appointment, it is entered in doctors schedule and patients records. Patients are able to rate a consultation visit experience with the doctor.

PHYSICIAN/NURSE/STAFF FEATURES

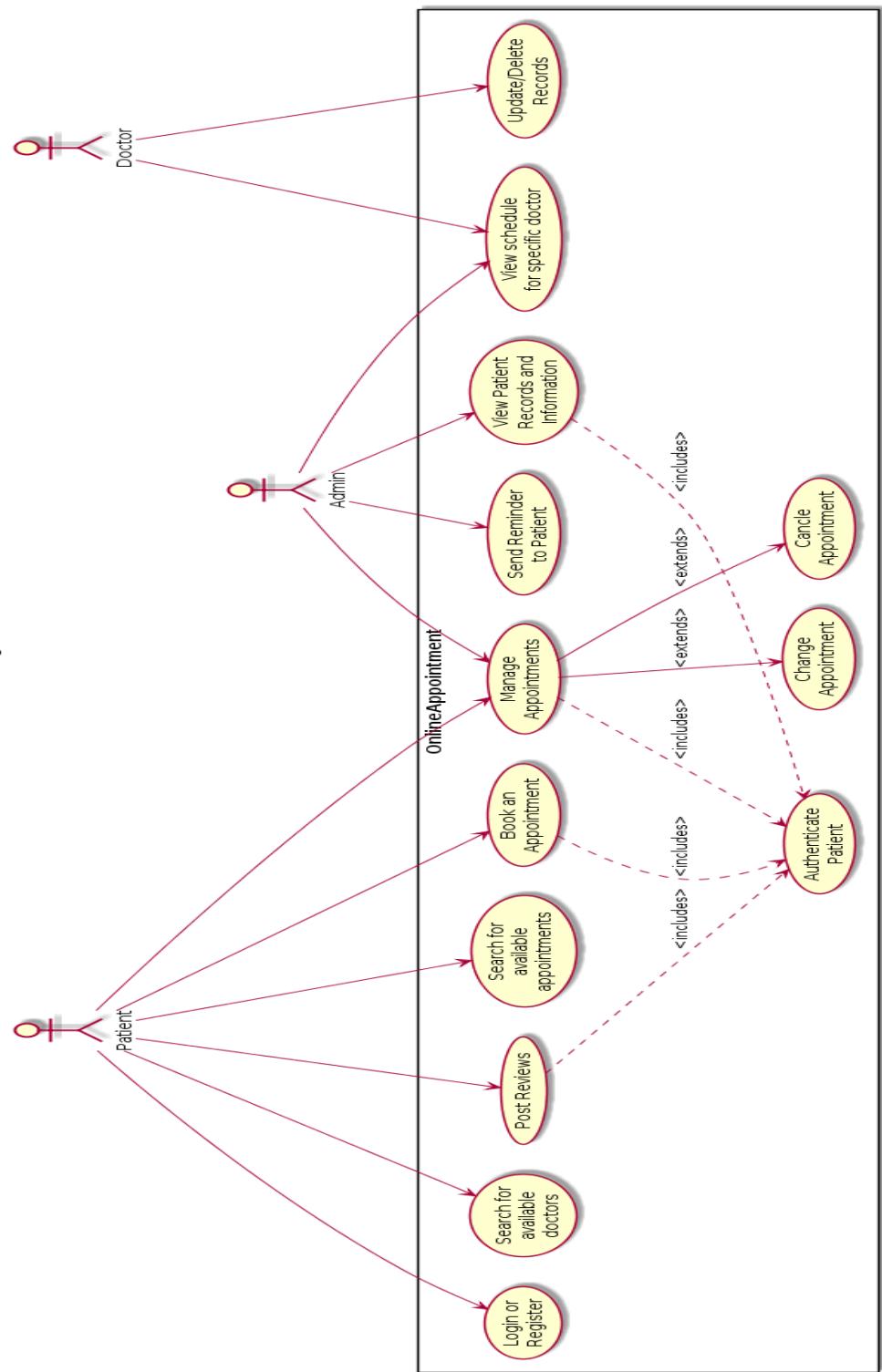
doctors would be able to view patient details. doctors could also keep track of no of patients that visit them. doctor could also update his availability status

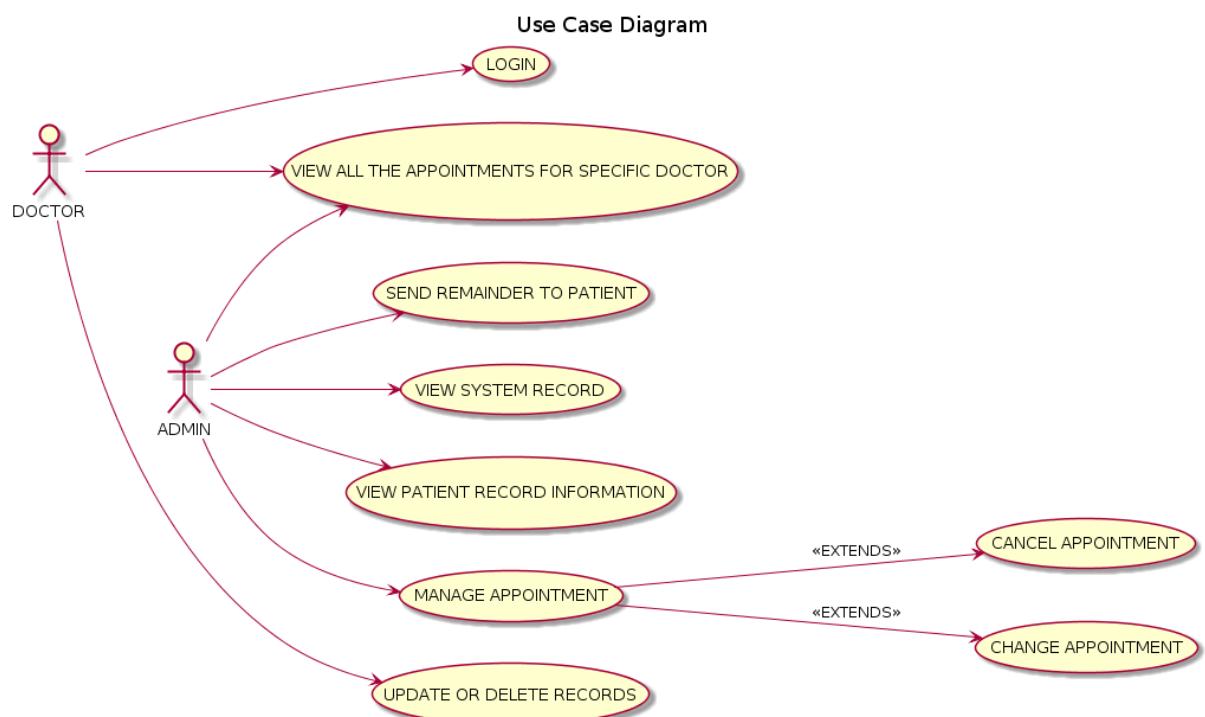
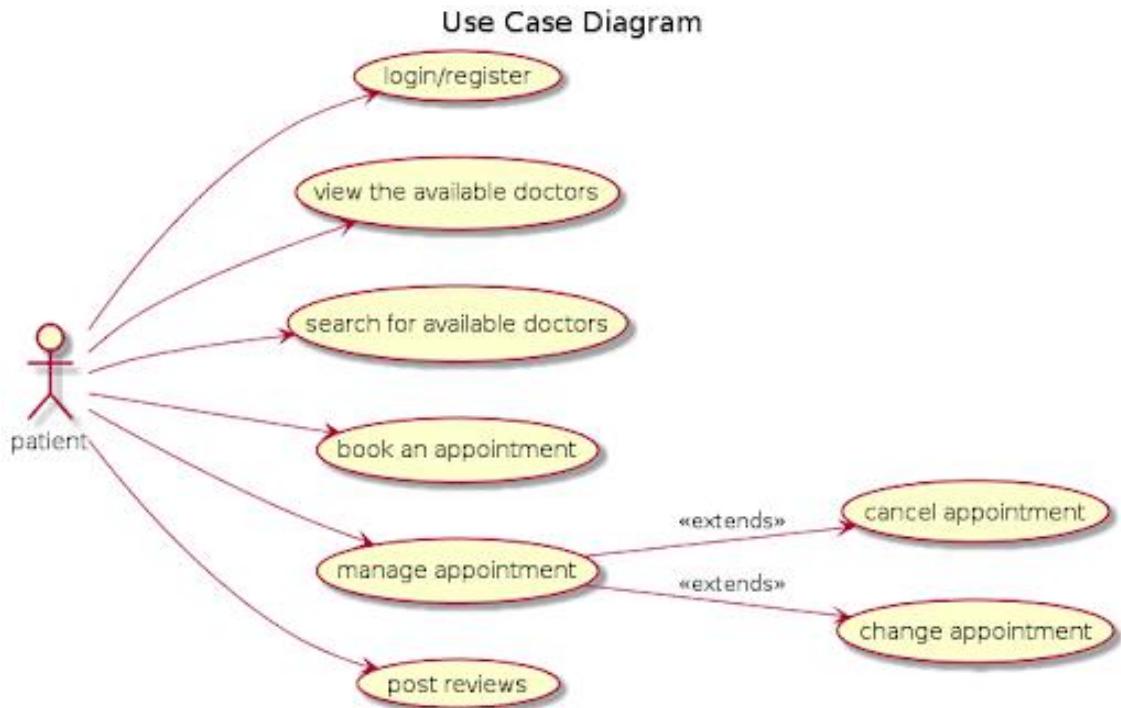
ADMIN FEATURES

Administrative staff would be allowed to register as new user in the System as Doctor,nurse,staff as well as control records (For example send message if doctor is unavailable due to some emergency)

OUR IMPLEMENTATION ABSTRACT IDEA TABLE:-

ONLINE PATIENT SCHEDULING SYSTEM		
1. Log-in	Admin User log in	
2. System Change Control (For Example: Adding new user to the system, updating the system)	Admin User Functions	1. Admin
3. Log-in, Register	Patient User Log in	
4. Department List		
5. Physician List		
6. Available Slots	Patient User functions	2. Patient
7. Book Appointment		
8. Make Deposit		
9. Update/ Cancel		
10. Log-in	Doctor User Log in	
11. Update Doctor Specification		
12. View Patient Detail	Doctor User Functions	3. Doctor
13. View Appointment Record		
14. Log in	Nurse, Staff User Log in	
15. View and Update Patient Detail		
16. View and Update Appointment Records	Management Staff User Functions	4. Staff





Software Requirement and Specifications

Beatsify

The Music Recommendation System

Koustav Chowdhury
17CS10059

Prakhar Sharma
17CS10037

What is the need?

In recent decade Rapid development of mobile devices and internet has made possible for us to have the scale of available tracks in hands rocketing from tens to millions. The number of songs available exceeds the listening capacity of single individual. People sometimes feel difficult to choose from millions of songs. Moreover, music service providers need an efficient way to manage songs and help their consumers to discover music by giving quality recommendation. Although there is a huge collection of music out there but only a small subset is generally recommended to consumers. Thus, there is a strong need of a good recommendation system.

How the need can be addressed?

The plan is to use Machine Learning for predicting the next few songs based on the current playlist of the user. The Million Song Dataset provided by Kaggle is a good starting point. The basic ambition of the project is the fine-tune suggestions based on the listening history of the user and plan is to further extend the prediction based on the people who listens to similar music which consumer listens.

Who will be the prospective users?

Anyone who loves music and is invested enough to tune his recommendation to suit his taste is a potential user. According to The Times of India , there are 200 million users on Gaana, and going by the recent trends, this industry is rapidly blooming in this country, thanks to the low-cost 4G network availability.

What are the issues/challenges to be overcome while developing the system.

The scarcity of data is a major concern. Even, if we manage to get past the initial stage of development, integrating with the existing apps like Gaana, Saavn, Soundcloud is another hiccup in the process of development.

Also developing carefully picked and handcrafted features like genre, artists and many more is also a tiresome work.

Plan the working of your project from today to till the end of the project.

The project as such is implementation-heavy, so it is wise enough to apply a bottom-up approach in developing the system.

Initially, the plan is to develop an web-app for deployment of our recommendation system which can play songs based on the user demand similar to a music player but the music will be used from digital platforms like Youtube, Spotify and etc.

Parallelly, the work on the Machine Learning part will also be going on. Scraping of playlist and learning based on the Million Song Dataset is the most fundamental work that can be done. Million Song Dataset provides many types of features for a song. We select eight features from the Million Song Dataset, i.e., loudness, pitches, tempo, duration, song hotness, artist similarity, artist hotness and artist familiarity, and these features cover the three categories. Furthermore, user preference is counted in too. The similarity is able to indirectly represent the performance of favorite song detection. Euclidean distance is applied to measure the distance of songs for features like terms, similar artists, artist hotness, artist familiarity, artist location, artist name and musicbrainz tags. Play count is considered as an implicit rating based on the assumption that the rating is positive if the song is

played many times, and vice versa.

Learning API programming should also be finished in the initial days. We plan to use APIs for playing songs so that we can apply real-time learning on the taste of the music.

There will also a login system for the user, to generate custom playlist based on the popular music played by other users. (This will be implemented at the last). There will also be a system to share your playlist with other users.

Functional requirements

□ Getting recommendation for songs

- This is the primary requirement of the user for which the system is being developed. The recommendation system takes log of previous songs as input and the output is a predicted song which will be generated by the system by applying learning on the dataset.

Development of Music Player

- Development of a Music Player is also a major requirement as without music player our music recommendation system is of no benefit.
- Music Player should have basic features like play, pause, forward, previous, repeat song and many more.

Non-functional requirements

Sharing of playlist

- This is a non functional requirement as it is not directly related to the recommendation system. Rather, it's like an add-on which is done for ushering user-friendliness.

Comparing music taste with your friends

- This is also a non functional requirement as it also is not directly related to the recommendation system. This feature will also act as an add-on and will serve as an icing on cake.

Tentative hardware/software environment to be used.

Web-app development and the database management will be deployed using Java Servlet and JDBC respectively.

Machine Learning aspect of the system will be dealt using python. We will be using Servers of Computer Science Department of IIT KGP for setup of deep and machine learning libraries. Libraries like Numpy, Tensorflow, keras will play major role in Machine Learning aspect.

**How would you like to publicize your system
(i.e., pointing out the attractive features, novel
things, etc.). Also, what would be the cost of
product according to your assessment.**

In today's world of hustle bustle where everyone is roaming around with their earphones plugged in, Music has become an essential part of everyone's life. Everyone likes to listen to music of their own taste and changing song every-time at the end of song is an irritating job.

So we propose a Music Recommender System with the most attractive feature of the system is its ability to recommend songs based on few songs played by the user and it also accounts many other features like user and his friends previous history for recommendation.

So why follow the crowd when you can get the music of your taste at your fingertips.

The cost of the product is expected to around 50k \$.

SOFTWARE REQUIREMENT SPECIFICATION DOCUMENT

Co-operative Bank

Rithin Manoj(17CS10043)
Venktesh Lagaskar(17CS30037)

Table of content

1. Project Description.....	1
2. Plan.....	2
2.1. Requirement Analysis.....	2
2.1.1. Functional Requirements.....	2
2.1.2. Nonfunctional Requirements.....	7
2.2. Hardware Requirements.....	8
2.3. Software Requirements.....	8
2.4. Cost and Production.....	8
3. Use case Diagram.....	9

1. Project Description:

Corporate banks are always places with high rush and the absence of employees can create a lot of problems for companies where each second matters and human errors can lead to wastage of time. This software will allow the companies and users to easily access all the policies provided by bank inline, enabling them to apply for something online, saving them a lot of trouble and it helps the bank too as all the data will be directly received by them, in a structured manner helping them to easily understand the query and help the user appropriately.

The project aims to address the issues by taking advantage of the astonishing reach of Internet around the globe nowadays. The need can be addressed by making a common platform, which can form partnerships with different banks enabling the user to choose his bank and access his account. This would enable them to request for loan or other query from other banks even though he don't have an account there.

The intended audiences for this document are the development team, the team manager, the customer and all other stakeholders in the system. The bank employees are also part of the expected audience

The Online service should be running 24x7 and should be accessible from anywhere. The main challenge is to maintain privacy for the customers and make sure that their information are secure. At the same time it should be reliable and maintained properly

2. Plan:

We start with SRS documentation, which involves project details, requirement analysis and hardware/software requirement and then proceed onto design of the system. After this comes the programming part, where the core programming is done followed by 3 steps, which are implementation, Testing and final submission respectively.

2.1 Requirement Analysis:

2.1.1. Functional Requirements:

a) Register:

INPUT - New user fills the details and sends the completed form

OUTPUT - The form is sent to the administrator.

PRECONDITION-The user should be a registered customer/ should register as a new user.

POST CONDITION- NA

New users want to start the online banking services they fill the form online and send to admin. Admin verifies that details and provides the service to users

b) Log-on Capabilities:

INPUT- The user name and password

OUTPUT- The personalized log-in page will be displayed.

PRECONDITION-the user should be a registered customer

POST CONDITION- NA

It is basically the log-in page through which user interacts with the Banking site.

2

c) Browsing:

INPUT- Clicks on the link to which the service belongs.

OUTPUT- The service of the banking web page view

PRECONDITION- The link pages should exist in the banking database.

POST CONDITION- NA.

The Online Banking can be providing the services to customers.

d)Print balance:

INPUT – NA

OUTPUT – Balance printed

PRECONDITION–NA

POST-CONDITION- NA.

Prints the balance in the users account

e)Money Deposit:

INPUT – Amount to be deposited

OUTPUT -success message and new balance

PRECONDITION– NA

POST CONDITION- NA.

Existing users deposit the money.

f)Money Transfer

INPUT – Amount to be deposited and destination account number

OUTPUT -success message and new balance

PRECONDITION–destination account number exists

POST CONDITION- NA

Existing users withdraw or transfer the money from directly or transfer to any other bank account

g)Loan Application:

INPUT – Amount of loan required, and required data
OUTPUT - Message, the bank's response to query. Yes/No
PRECONDITION– NA
POST CONDITION- NA.

Responds to the loan query.

h) **Update Balance:**

INPUT – Withdraw/deposit
OUTPUT - Message, the balance
PRECONDITION– User has done an action
POST CONDITION- NA.

Responds to the balance query.

i) **Find Customer:**

INPUT – N.A
OUTPUT - Message, the list of all users with their Account Numbers
PRECONDITION– User is Administrator or Bank
POST CONDITION- NA

Allows to withdraw money

j)Complaint:

INPUT – The Complaint

OUTPUT -N.A

PRECONDITION– User is Customer

POST CONDITION- N.A

Allows to post complaints

k)Respond to complaint:

INPUT – Reply to complaint

OUTPUT -N.A

PRECONDITION– User is Bank

POST CONDITION- N.A

Allows to view and respond to complaints

l)Request for cancellation:

INPUT – Request to cancel account

OUTPUT -N.A

PRECONDITION– User is Customer and balance is zero

POST CONDITION- N.A

Allows to request cancellation of account

m)Cancel:

INPUT – N.A

OUTPUT -N.A

PRECONDITION– User is Bank and a request is received

POST CONDITION- Account cease to exist

Allows to cancel account

n)Customer Information:

INPUT – N.A

OUTPUT -Customer Information

PRECONDITION– User is Customer

POST CONDITION- N.A

Allows to view the customer information

a)Usability:

The users of the system are members and the administrators who maintain the system. The members are assumed to have basic knowledge of the computers and Internet browsing.

b)Reliability:

The system has to be very reliable due to the importance of data and the damages incorrect or incomplete data can do.

c)Security:

There shall be no security mechanisms in place to keep unwanted users out of the system. However, users of the system shall not be able to perform actions or request actions for other users,due to password protection

d)Availability:

The system should be available as long as internet and a device(mobile, pc, tab etc.) is available.

e)Maintainability:

We should be able to maintain it really easily and should also be able to add code easily in case of upgrades

2.2 Hardware Requirement

- 1)Standard PC
- 2)Processor with Integrated GPU

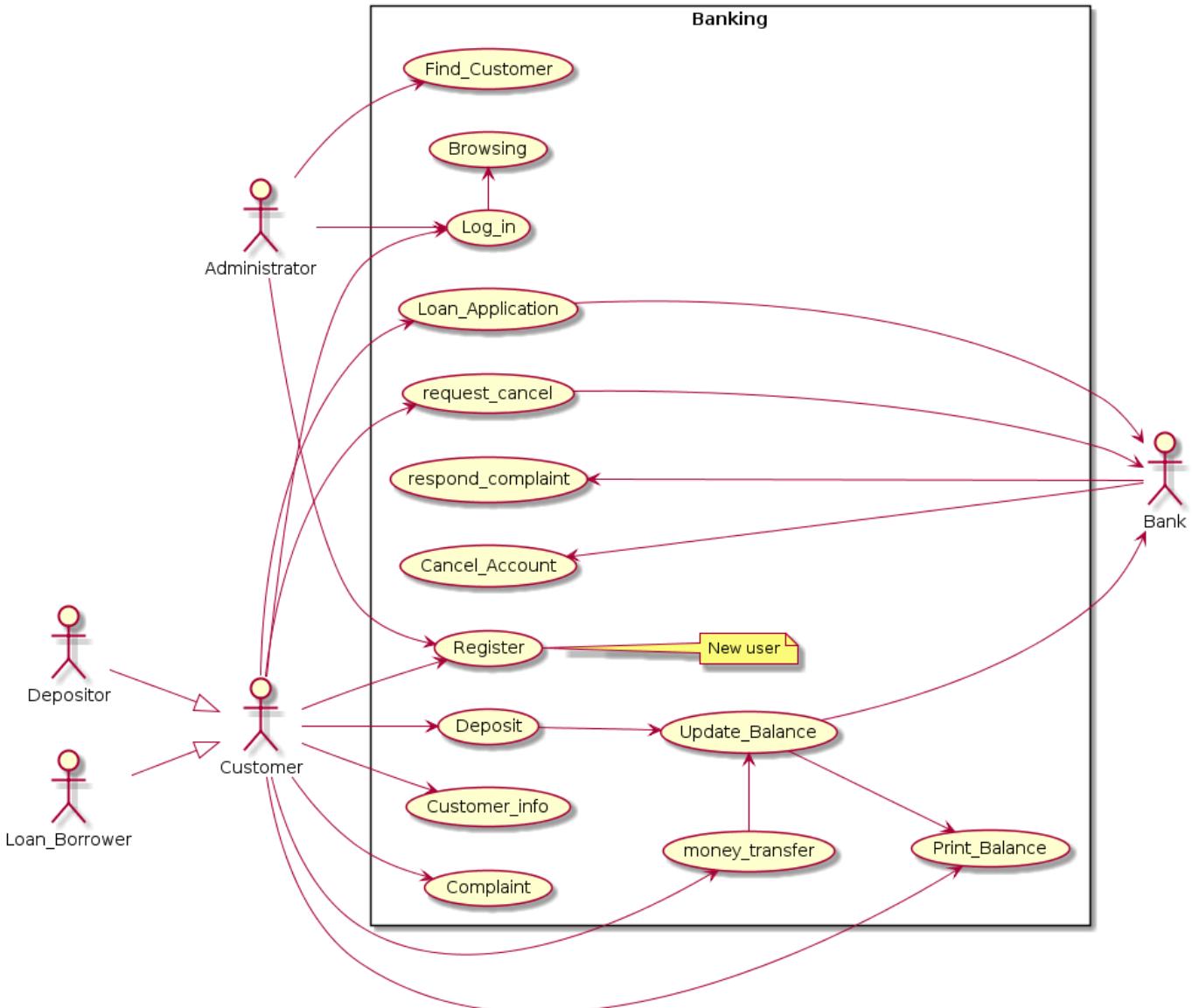
2.3 Software Requirement

- 1)Java
- 2)Python
- 3)Libraries for machine learning

2.4 Cost and Production

The system is supposed to be free of cost and will be publicized online for free

3. Use Case Diagram



Software Requirements Specification

Digital Signature Software

Version 1.0 approved

Prepared by:

Aaryan Bhagat (17CS10001)

Faraaz Rahman Mallick (17CS30043)

Group - 22

8/2/2019

Table of Contents

Table of Contents	ii
1. Introduction	1
1.1 About Digital Signature	1
1.2 Need of the Software	1
1.3 How the need can be addressed	2
1.4 Prospective Users	2
1.5 Challenges/Problems to the Development Procedure	3
2. Project Planning	4
3. Functional Requirements	5
3.1 Hash Generation Function	5
3.2 File Send Function	5
3.3 Signature Algorithm Function	6
3.4 Verification Algorithm Function	6
4. Non Functional Requirements	7
4.1 Performance	7
4.2 Security	7
4.3 Backup	7
4.4 Portability	7
4.5 Usability	7
5. Tentative Hardware/Software Environments	8
5.1 Hardware Requirements	8
5.2 Software Requirements	8
6. Publicity and Expected Expenditure.....	9
6.1 Publicity Measures.....	9
6.2 Expected Cost.....	9

1. Introduction

1.1 About Digital Signature

A Digital Signature is a Private Key, which can be obtained from a Certifying Authority after submitting your Documents. It is the hash stored in the hardware and that can be issued with a public key and then that can be used to sign the documents. It is a type of Mathematical Authentication Model used for the authenticity of digital messages or documents. A valid Digital Signature gives the recipient a reason to believe that the message was created by a known sender and it was not altered, in transit.

1.2 Need for the Software

There are basically three needs for a digital signature:

- a) Integrity
- b) Authentication
- c) Non-Repudiation

Protecting the signature at the point of signing :

A digital signature is an encrypted file that travels with the electronic document that needs to be signed and returns with it after the transaction has been completed. The file contains and captures the complete meta-data about the file (e.g. where it travelled to, who opened it, the precise timing of interaction etc.)

Protecting the signature in storage:

A digital signature prevents the electronic file from being tampered with. It becomes very easy to verify whether the signature has been tampered with or not, over its period of storage. Thus, it acts as an electronic seal.

Protecting the signature on mobile devices:

Smartphones and tablets are the latest contenders in the field of technology. Naturally, people want to use their mobile devices to do work, which, many times, the devices are not built, or configured to do, e.g. they do not feature the same level of security and data protection features which are present in traditional PCs. The trick is, we can CONFIGURE a digital signature to gather more critical meta-data in a customizable format.

1.3 How the need can be addressed?

The need can be addressed by developing an easy to use, efficient, portable and flexible software, which can easily allot a digital signature to any document, which needs to be transferred securely and uses minimal hardware and fewer resources. It should also be able to prevent the document from digital interception (which is the access of files by unauthorized individuals while it is being transferred over communication lines). It should also be able to help in the verification of documents on the receiver's end so that he knows that he has received the file from the correct individual.

1.4 Prospective Users

A Digital Signature is something which is very important in all fields of life. Be it a transfer of secure files in a school/college, or be it the transfer of highly classified files between various military agencies or between two governments, digital signature plays a very important role in facilitating the secure transfer of documents.

Some broad fields which use Digital Signatures:

- a) *Architecture, Engineering and Construction (AEC)*
- b) *Healthcare and Life Sciences*
- c) *Financial Services and Insurance*
- d) *Governments and their departments*
- e) *The Armed Forces (for transfer of sensitive information)*

In general, most global-scale companies which require transmission of important files need a method to verify whether the received one has not been tampered with or not. Ignorance in this area can lead to leakage of data of the organization and also much more apart from huge financial losses.

Therefore, any user looking for a method of safe data transfer over an unsecured channel to a peer will be in dire need of this type of tool.

1.5 Challenges/Problems to the Development Procedure

The challenges faced here are not trivial:

Generation of a unique digital signature:

A method of generating a unique id (“signature”) is required. It is the most crucial step we need to ensure that there does not exist two files having the same signature as this will ensure no collision of the hash functions. Hash collision can lead to the receiver obtaining a wrong file even though the signature is verified.

Interface for file transfer:

An interface is required for the secured transmission of files over the network. The type of network does not matter as long as it is secure. The GUI should be user-friendly and should support the latest GUI features and tools.

Flexibility and portability:

The software should be lightweight as more memory consumption would lead to slower processing and thus unsatisfactory results. It should be platform independent so that it can be used by anyone regardless of the OS of his system.

Debug Console:

A debug console should be present for customers who are developers as well. This can lead to easy identification of errors (bugs) if any, during the alpha and beta testing of the software. It would also lead to satisfaction among the customers as they can also contribute to the source code and make it better.

2. Project Planning

In simple terms we would try to stick to the following weekly plan:

Week 0: Requirement Analysis and Specification and Feasibility Study:

Here we would discuss what our software should do, and how we should go about developing it.

Week 1: GUI Programming (Preferably using Java Swing)

Here we would develop the basic GUI of the software so that it becomes easy to use by the masses.

Week 2: System Analysis and Design

Here we would analyze the system and try designing the various components (e.g. a function for hashing, a function for key matching etc.)

Week 3: Networking

This is one of the most important weeks, as we would now enable the program to be able to securely transmit and receive files over a network, without any issues (hopefully!).

Week 4: Implementation and Testing

We would implement all the methods and incorporate them to the main body of the program. Then we would test the software in real-time, by actually digitally signing a file, sending it over a network, and verifying its authenticity on the terminal of the recipient.

Week 5: Debugging

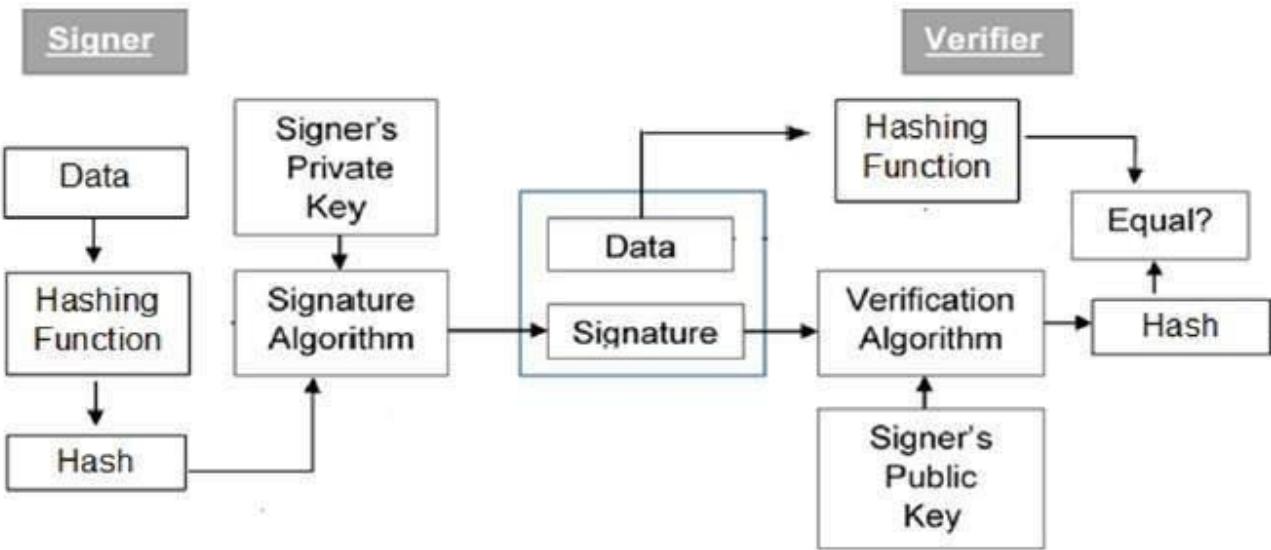
We would debug the software if we find any flaws, and correct them.

Week 6: Final Documentation

We would document all the characteristics and uses of the software along with the changes to the software after debugging, and provide a final SRS Document, which is flexible.

NOTE: This is an ideal work plan. It is subject to extensions/changes by a week or two, depending on the circumstances.

3. Functional Requirements



The major functional requirements are:

3.1 Hash Generation Function

This function is a crucial part of our whole software. It will take the file as the input and will return the public and private key pair. The private key is kept in secret and only known by the sender of the file. The public key is distributed to the audience for verification of the signatures. Most in-built java packages ("java signature") are able to implement this function and is state of the art.

Input: The file (data) to be transferred.

Output: A unique hash.

3.2 File Send Function

This function will use the existing network interface of the system for the sharing of files over the network with the "digital signature" so the recipient can confirm the authenticity of the file with the signature and the public key he has. This function should work smoothly, have a minimal response time and should provide detailed logs for error identification during network errors.

Input: The file which is to be transferred, along with the address of the receiver.

Output: A message saying the status of the file.

3.3 Signature Algorithm Function

This function will take the private key and the hash generated as the input (refer to the flow chart).

It will return a “signature” which will be sent along with the original file, to the recipient for verification of its authenticity.

Input: The hash generated by the Hash Generation Function, and the sender’s private key.

Output: A unique signature.

3.4 Verification Algorithm Function

This function will take the public key of the sender (available to the recipient) and the digital signature of the file received to generate a hash which it will then compare to the hash of the file received (generated by the recipient’s hash function) to check the authenticity of the file received.

Input: The sender’s public key and the signature of the file received.

Output: Verification status of the file received (Verified or Not).

4. Nonfunctional Requirements

4.1 Performance Requirements

The software should be lightweight and should not consume excessive processing power. It should have a high response speed with a minimal lag present, i.e. efficiency should be supported with resource utilization.

4.2 Security Requirements

Software should contain the latest dependencies of the requirements so that the major vulnerabilities should be patched and no exploitation should take place. In other words, it should be prone to hacking attempts.

4.3 Backup Requirements

The software should be equipped with a back-up facility so that loss of data is prevented in case of impromptu shutdowns or other errors, leading to the crashing of the software.

4.4 Portability Requirements

The software should be usable in different environments and the pre-requirements for different environments are minimal and almost the same.

4.5 Usability Requirements

The software should have a user-friendly interface and the tools should be easy to comprehend and use. Most of the complex work should be done in the back-end away from the screen, which would lead to easy handling and thus customer satisfaction.

5. Tentative Hardware/Software environments to be used

5.1 Hardware Requirements

Minimal hardware is required:

- 1) A portable/desktop machine with any Operating System (preferably, the latest version)
- 2) The machine should have an average processing power as per today's standards (which is > 1.0 GHz clock/cycle)
- 3) Working input and output devices.
- 4) A decent, working network connection which allows file transfer, and related routing devices.

5.2 Software Requirements

- 1) A programming language (preferably Java) and all related packages for easy compilation and building the software.
- 2) A lightweight Graphical User Interface toolkit (preferably Java Swing, as it is platform independent, and easy to use)
- 3) Drivers for facilitating the transfer of files over a network, securely.
- 4) For networking, we would use the Java.net package (which has different modules like ServerSocket, Socket, InetAddress etc).

6. Publicity and expected expenditure

6.1 Publicity Measures

The dire need for such type of software tool in the market itself helps in its publicity. The target audience is diverse and the urgency of this tool is utmost in today's connected world. Key points for publicizing our software are as follows:

Our model will provide:

- 1) Best state-of-the-art encryption for the files for maximum security.*
- 2) Minimal time in hash-generation and no lag in response, even if the file size is large and the number of requests to the server is more than average.*
- 3) Easy to use and comfortable interface which is platform independent for reaching out to a wider audience.*

6.2 Expected Cost

Since this software is the solution to a critical problem (which is of the secure transfer of files over a network), we would like to keep this software as an open-source project, which would be free of cost, and anyone with sufficient knowledge can contribute to the source code so that even more functionalities could be added to an already good software.

SOFTWARE REQUIREMENTS SPECIFICATION

FOR

ADVANCED CALCULATOR

Version 1.0

Prepared by Soumen Dutta (17CS10057)

& Anshul Choudhary (17CS10005)

Indian Institute of Technology, Kharagpur

13 Feb 2019

Table of Contents

Contents

1. INTRODUCTION.....	3
1.1 Purpose.....	3
1.2 Product Scope.....	3
1.3 Challenges	3
2. Overall Description.....	4
2.1 Product Perspective.....	4
2.2 Product Functions.....	4
2.3 User Classes and Characteristics	4
2.4 Operating Environment.....	4
3. Tentative Timeline	5
4. Functional Requirements.....	5
4.1 User Interface Requirements	5
4.2 System Features.....	5
5. Nonfunctional Requirements	6
5.1 Performance.....	6
5.2 Reliability.....	6
5.3 Usability	7

Revision History

Name	Date	Reason For Changes	Version
Advanced Calculator	13 Feb 2019	-	Version 1

1. INTRODUCTION

1.1 Purpose

The purpose of this document is to present a detailed description of the Advanced Calculator Application. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. The application is designed to act like a handheld scientific calculator with the usual standard functions like add, subtract, multiply, divide and memory. Additionally, it will have the capability of computing trigonometric functions, exponential functions, differentials, integrals, matrix calculation, etc. The program is designed to be user friendly.

1.2 Product Scope

This software is a scientific calculator which have more features as compared to the ordinary calculator. The functions which are involved in this type of calculator are scientific notation, floating point values, logarithmic functions, exponential functions, complex numbers, fractions, equation solving, matrix calculations, calculus, conversion of units and physical constants. The scientific calculator is used in various fields for example in astronomy, geology, physics, chemistry, somehow in biology as well for performing the basic functions as well as complex calculations. This software will be useful for students, scientists, and nearly anyone who performs arithmetic in his daily life. The goal of this application is to provide the users with a one stop solution to all their computation needs. The application will be designed to maximize the productivity of students and minimize the time required for mathematical computation.

1.3 Challenges

Designing an Advanced Calculator require a good understanding of higher mathematics as well as programming language. Simple arithmetic programs are not a big task but performing calculus and matrix algebra might pose a problem. For large calculations, calculator may take too much of time, so the implementation must be efficient. The chief challenge is the implementation of all the above mentioned functions in an optimal manner. As for several of these functions are not available in default java packages, so we may need to use other open source packages for the same.

Apart from these the other challenges that we need to overcome include developing a user friendly GUI design and effective exception handling in case of erroneous input.

2. Overall Description

2.1 Product Perspective

Advanced Calculator is standalone desktop application that can run on Windows and Linux Platforms. This software is written java programming language and thus will require at least JRE (Java Runtime Environment) for execution. Apart from that it is independent of any other application.

2.2 Product Functions

The key features of advanced calculator include:

- Scientific notation
- Floating point arithmetic
- Logarithmic functions, using both base 10 and base e
- Trigonometric functions & hyperbolic functions
- Exponential functions and roots beyond the square root
- Quick access to constants such as pi and e
- Complex numbers
- Fractions calculations
- Equation solving
- Matrix calculations
- Calculus
- Variable creation and storage
- Conversion of units
- Physical constants

2.3 User Classes and Characteristics

The principle users of this product are considered to be students pursuing higher education and professionals who require a scientific calculator for their work. It is considered that the user does have the basic knowledge of operating a desktop computer. Also he/she must have a basic knowledge of mathematics to able to use this product.

2.4 Operating Environment

The software shall be written in standard Java8. It will use javac as compiler (for conversion from source code to Byte code) and JVM interpreter (to run the byte code).

The software shall use standard java packages as well as other open source packages required for calculus and matrix algebra. The application is designed to work in a desktop environment that supports JVM.

3. Tentative Timeline

- **Week 1:** Creating and Finalizing SRS Document
- **Week 2:** Coding Basic Arithmetic calculation module
- **Week 3:** Coding Exponential and Trigonometric module
- **Week 4:** Coding Calculus module
- **Week 5:** Coding Unit Converter and Equation solver modules
- **Week 6:** GUI Development (Day - I)
- **Week 7:** GUI Development (Day - II)
- **Week 8:** Testing, debugging and module integration

4. Functional Requirements

4.1 User Interface Requirements

The calculator in default mode has the following keys: 0/x, 1/y, 2/z, 3/pi, 4/e, 5, 6, 7, 8, 9, ., +, -, *, /, ±, ^, %, =, (,), C, CE, shift and 1-operand keys: 1/x/mod, x2/sqrt, sin/argsin, cos/argcos, tan/argtan, exp/ln, 10^x/log, mode, ans, gcd/lcm, nPr/nCr, var.

In any situation the calculator has to produce a correct result defined by the well-known arithmetic rules. If the calculations are impossible the calculator has to display information helping the user to resolve the erroneous situation, like:

- On encountering a division by 0 the display should read "Cannot divide by 0" and typing the key "C" should reset the calculator.
- On calculating the square root value of a negative operand the display should read "Wrong operand".
- On erroneous operand or operation keys the display should read "Reset (C) to continue" or "Clear (CE) to continue" as appropriate. Of course, any situation can be cleared using the main reset key "C".

4.2 System Features

- **Input**

The standard calculator input will include a string composed of numerical symbols, operators, functions (trigonometric, exponential, etc.), variables, etc. which will be evaluated according to standard mathematical convention.

- **Basic Calculations**

The string previously taken as input will be processed. First all the trigonometric and exponential functions will be evaluated separately and the value will be returned to the Calculation function which will then perform the basic mathematical operation (+, *, -, /) according to the usual precedence rules.

- **Integration**

In this mode user can compute the definite integral of a one variable function. This mode will have a separate interface showing three fields for upper, lower limits and the function to be integrated in terms of variable X. The output after computation will be displayed in the separate field.

- **Differentiation**

In this mode user can compute the derivative of a one variable function at a particular on the real line. This mode will have a separate interface showing two fields for the function to be differentiated in terms of variable X and the point at which the derivative is to be evaluated. The output after computation will be displayed in the separate field.

- **Matrix Algebra**

In this mode user can compute the product, sum and difference of two matrices. The size of the matrix (M_{ij}) can vary between $1 < i < 4$ and $1 < j < 4$.

- **Equation Solver**

In this mode user can solve linear/quadratic/cubic equation of 1 variable and set of multivariate linear equations up to three variables.

5. Nonfunctional Requirements

5.1 Performance

The system must be interactive and the delays involved must be less. While performing large calculations, the system must not take more than 2 seconds to output the result. This can be ensured using efficient algorithms and JAVA libraries.

5.2 Reliability

As the system provide the right tools for problem solving it must be made sure that the system is reliable in its operations and for giving correct results

5.3 Usability

The system should be easy to handle and should navigate in the most expected way with no delays. It contains all the necessary operations and some advance operations as well, which increases the usability of the program.

Software Requirements Specification

for

VOTE

Version 1.0 approved

prepared by

Prabhpreet Singh Sodhi
Indian Institute of Technology, Kharagpur

Sahil Singh
Indian Institute of Technology, Kharagpur

8th February, 2019

Table of Contents:

1.1 Category

1.2 Purpose

1.3 Scope

1.4 Introduction

1.4.1 Existing System

1.4.2 Proposed System

1.5 Advantages

1.5.1 Benefits to Elected Representatives

1.5.2 Benefits to people

1.5.3 Benefits to contractors

1.6 Requirements

1.6.1 Functional Requirements

1.6.2 Non-functional Requirements

1.7 Software Tools

1.8 Deployment

1.9 Hardware Specifications

1.10 Economics

1.11 Marketing

1.12 Distribution

1.13 Proposed Plan of Action

1.14 Challenges to overcome

1.15 Limitations and Future Scope

Project Title-
VOTE: Voice of the Electorate

1.1 Category: Android Application

1.2 Purpose:

The purpose of developing this application is to provide a common virtual platform to the people for voicing out their concerns directly to the elected representatives of the concerned constituency, and for their efficient and cost-effective solution by the best bidder.

1.3 Scope:

The aim of this application is to facilitate the effective solution of the day-to-day problems faced by the people, which often see a delayed response by the authorities, by bringing the elected representatives and the people, along with the contractors interested in solving the issue to a common platform. It can be implemented in all democratically functioning states, for providing a transparent and active platform of grievance removal.

1.4 Introduction:

The utility of this application can be best understood by comparing the existing system with the proposed system:

1.4.1 Existing System:

The process of getting even minor public works done by the authorities is relatively cumbersome and time consuming, which generally involves sufficient manual intervention from lodging the complaint to getting the job done. Further,

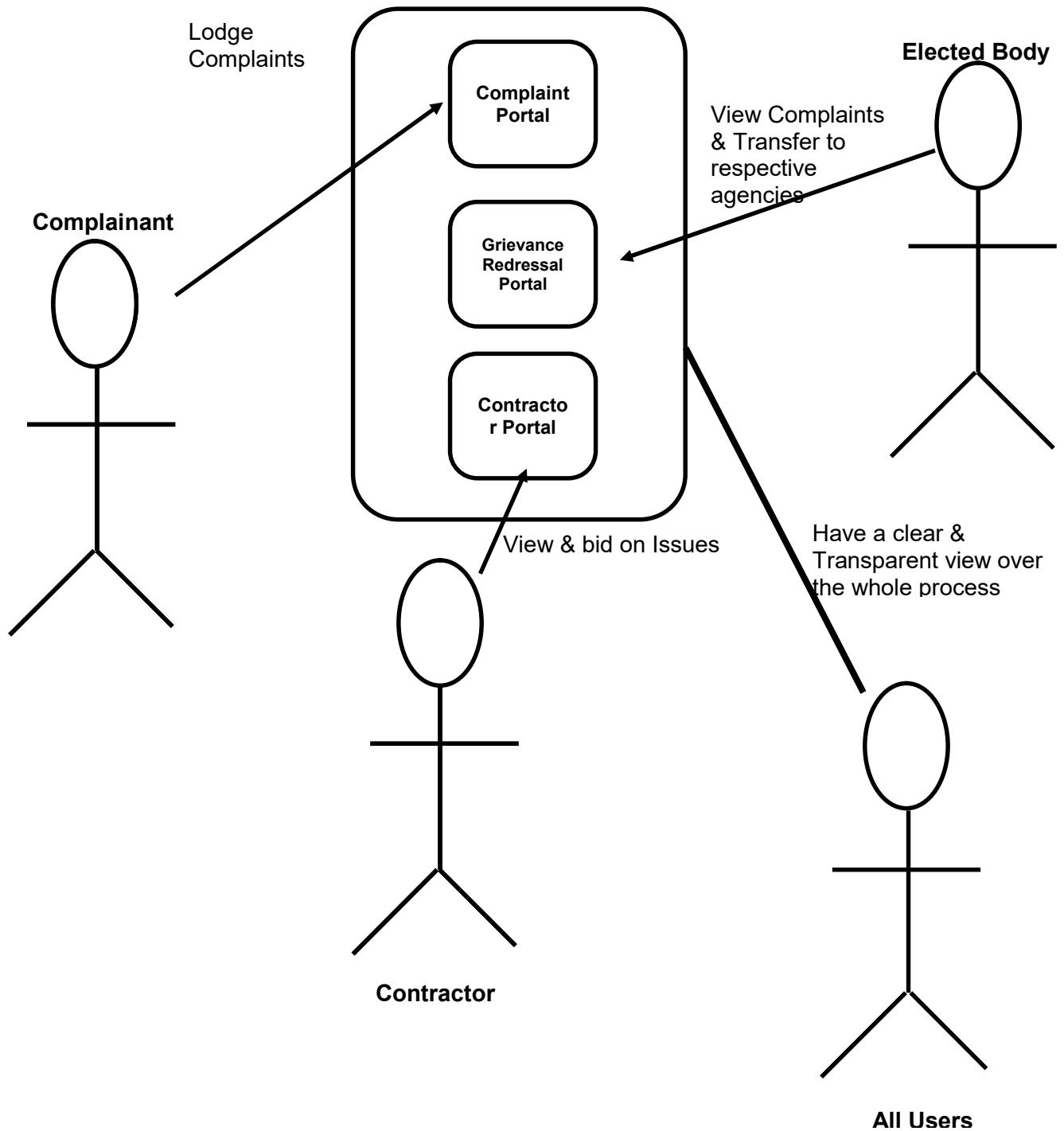
there is practically no proper method to make the concerned authorities aware of the number of people getting affected by a particular issue i.e. the scale of the problem. Besides, there is no proper quantitative measure available to the people for measuring the performance and response-time of their elected representatives at large.

1.4.2 Proposed System:

Our goal is to make the process of grievance removal more transparent and less cumbersome, while simultaneously saving on time required to lodge such complaints.

After careful analysis of the scenario, we present the following modules in our application:

- **Representative Registration:** Constituency-wise registration of the elected representatives.
- **Electorate Registration:** Constituency-wise registration of the voters of the area.
- **Grievance Portal:** Members of the electorate can lodge their complaints by choosing the respective categories of sub-area, domain of complaint (water, electricity, etc.), severity, etc. Similar complaints will be merged together to avoid duplicity.
- **Redressal Portal:** The concerned representative can view all the complaints lodged so far, and take appropriate actions.
- **MP/MLA Report Card:** It will display the fraction of issues solved by the concerned representative, along-with other parameters of performance such as average time taken in solving an issue, etc.
- **Contractor Portal:** It will include a list of registered contractors, which can make bids to solve any given issue. The most competent bid can be made to work on the same, thereby reducing the pressure on government agencies.



Diagrammatic Illustration

1.5 Advantages:

The proposed system offers several state-of-the-art benefits and improvisations to the way public affairs are handled.

1.5.1 Benefits to Elected Representatives

- Better management of issues concerning the constituency
- Better and cost-effective management of funds and grants
- Better identification of under-developed regions of his/her constituency

1.5.1 Benefits to People

- Timely removal of grievances
- Objective measure of competence of various representatives
- Hassle-free system to get public works done
- Transparency in expenditure of government funds

1.5.2 Benefits to Contractors

- Increased scope of business
- No long delays in granting of tenders
- Corruption-free and transparent business environment
- Can contribute productively to the common welfare of the people

1.6 Requirements:

1.6.1 Functional Requirements

➤ Login and Registration Portal:

- Allotment of username on the basis of Voter ID to avoid duplicate accounts.
- Password Encryption to avoid stealing of passwords.
- Each voter and representative can be registered from at most one constituency.
- Contractors to be given UIDs (Unique Identifiers) to avoid fake tenders.
- **Input:** Personal details such as Name, Address, VoterID, etc.
- **Output:** Unique Username for future login

➤ Complaint Portal (for People):

- New issue portal:
 - Mechanism to lodge new complaints
 - **Input:** Details about the problem such as sub-division, locality, domain and severity, etc.
 - **Output:** Complaint ID, and updated list of complaints
- Existing issue portal:
 - Mechanism to view, upvote and remark on an already raised issue
 - Mechanism to merge similar issues together to avoid duplicate complaints
 - **Input:** Actions such as upvote and comment on an already existing issue, or details about the problem if the user is creating a duplicate complaint
 - **Output:** Updated parameters of the issue, directly or indirectly by merging the duplicate complaint with the original one

➤ ***Grievance Redressal Portal (for government representative):***

- View Complaints:
 - Sort complaints according to domain and severity
 - Obtain details of complaints being raised in different areas of the constituency
- Transfer of Issue to Contractor:
 - Representative can choose the best and most competent contractor to transfer the issue for its resolution.
 - **Input:** The chosen contractor
 - **Output:** Status update on the issue with details of contractor and estimated time
- Transfer of Issue to Government Agencies:
 - Mechanism to transfer the issue to government agency if no contractor makes competent bid
 - **Input:** The chosen agency
 - **Output:** Status update on the issue with additional details such as estimated time
- Notifications and Broadcast Tab:
 - To view notifications from associated electorate
 - Broadcast important messages effectively to masses
 - Create polls to know the plan of action most preferable to the masses
 - **Input:** Message to be broadcasted
 - **Output:** Message or poll notification is broadcasted to all users

➤ ***Contractor Portal:***

- Make Bid Portal:
 - Choose an issue to bid on
 - **Input:** Issue chosen, estimated cost and time

- **Output:** Details sent to Representative
- Confirmation Portal:
 - Confirm tender if chosen for a job
 - **Input:** Confirmation Details
 - **Output:** Issue status updated

➤ ***General Features:***

- Representative report card:
 - Display various performance parameters of the representative giving a measure of how the issues are being handled
 - Can be viewed by all
 - **Input:** None (Data is collected with every issue raised automatically)
 - **Output:** Performance card of the representative
- Contractor rating:
 - Voters can rate contractors based on the quality of work done in solving the issue
 - Too many negative feedbacks can lead to suspension of account of the contractor
- To facilitate transparent utilization of funds, amount bid by the contractors will be visible to the people in the constituency
- Behavioral code of conduct to promote healthy environment.
Violators will be warned and ultimately banned.

1.6.2 Non-functional requirements

- Secure storage and access of sensitive data such as Voter ID number, etc. to ensure the data security of the people enrolled.
- Better component design to get maximum efficiency, to ensure the ability of the system to run on a wide variety of platforms and hardware environments.
- System must be interactive to enable it to be used effectively by the common masses.
- Lags in performance must be minimized to boost efficacy of the system as a whole.

1.7 Software Tools:

- Android Studio
- Java
- Python 3
- MongoDB/MySQL

1.8 Deployment:

- Any device with Android operating system

1.9 Hardware Used:

- A Personal Computer with Windows/Linux OS
- GPU

1.10 Economics:

Since the development of this project does not require any special hardware, and is being developed solely for learning purposes, it will be supplied at no cost.

On a personal note, we specifically chose this problem statement to build a system which can be put to practical use for the welfare of the country, and hence, it will be made open-source after the development and testing phase.

1.11 Marketing

We have listed the numerous advantages the software offers in comparison to the existing non-digital solution, which we believe are sufficient to draw the attention of the masses. Further, as the software will be a free of cost open-source application, it can be put to use anywhere and everywhere it is found useful without any permission from the developers. Due to economic restraints, we have not planned any specific commercialized marketing plan as of yet.

But, informing various levels of government about its utility, describing the various novel features (already mentioned) it offers, is surely on the cards.

1.12 Distribution:

We can use various digital application distribution services such as Google Play Store, Amazon App Store, Mobogenie Market, etc. for its effective distribution.

1.13 Proposed Plan of Action:

	WEEK 1	WEEK 2	WEEK 3	WEEK 4	WEEK 5	WEEK 6
Task No.	01/03 to 08/03	08/03 to 15/03	15/03 to 22/03	22/03 to 29/03	29/04 to 05/04	05/04 to 12/04
Task Details	T1	T2	T3	T4	T5	T6
Design Login and Registration Portal	Design Complaint Portal	Design Grievance Redressal Portal	Design Contractor Portal	Link all 3 portals together	Testing Phase	
None	T1	T1	T1	T1	T2, T3, T4	

Final Submission: 12/04/2019

1.14 Challenges in Development:

- To make a light-weight application that works well without a lot of hardware and processing requirements, for better penetrability
- Make the application interface simple enough to be understood by the less educated masses
- Work in accordance with the planned schedule

1.15 Limitations and Future Plans:

➤ ***Local Language Support:***

Since a large section of Indians cannot communicate or understand English, especially in the rural areas, the system cannot be put to efficient use unless it is made available in different languages spoken across the country such as Hindi, Bengali, Telugu, Tamil, Punjabi, Marathi, Gujarati, etc.

➤ ***Voice control:***

Since a considerable section of our population is illiterate, extending this system to provide full-fledged voice navigation is on the cards.

➤ ***Approval:***

The project cannot be put to effective use unless the government representatives approve of this as an official mechanism of grievance redressal.

➤ ***Image Processing based voter information retrieval:***

Considering a major chunk of our population is illiterate, for account creation, an image processing based solution is on the cards, where a user can simply upload a soft-copy of his/her voter card, for account creation instead of manually entering all the details.

Software Requirements Specification

Title : <Online Tic-Tac-toe Game>

Prepared by <Jeethender Naik Gugulothu>

Prepared by <Jayanth Harsha Gorantla>

<IIT KGP Students>

<8-02-2019>

1. About the Project :

►The basic need is to allow people to play tic tac toe online irrespective of their location. It should be accessed very easily and user friendly.

►Once connected to the server, we have a two way pipeline through which both client and server can send data to each other. The main advantage of this is that the server can send data to the client without the client even requesting it! This is particularly helpful for scenarios where we need to push data to the client based on some events.

►We'll be creating our own events to help build our game. Here is a brief overview of how our game will work. One player will create the game (Player 1 or X) and will provide the other player with a game Id. The second player will join the game (Player 2 or O).This game has alternating mode which ensures no discrepancy .This game will exist till we have a winner or the game is tied.



Model Diagram of Tic-Tac-Toe

2. Need for online gaming :

- » Now a days many people are involved in their own works and don't find relaxation to their mind. So, by this game, people get relaxed and some feel nostalgic while playing with their friends who are very far from them. Hence, there is a great demand of online platform for gaming which helps distant people to connect easily and enjoy the gaming. Therefore it reduces the boredom of the people and also helps in communicating with distant friends with added benefits like chat box .
- » This is also helpful for those who enjoys competition for developing their skills.

3. How the need can be addressed?

- » Major issue is to create an online platform in which the players can login and play the game.
- » First and foremost thing to do is to create a login page which consists of username and password etc.
- » We will connect the users through servers by creating game id .
- » We create a chat box for the users while gaming.
- » We can also create levels for users interest and help box for new users.
- » We can provide daily challenges for bringing interest in people.
- » The game should be able to work on all platforms. It must coexist peacefully with the music system which provides game sounds. It should also support a help system that provides instructions or hints to players.

4. Prospective users :

- Users are at various levels like who play for entertainment, play for competence to develop their gaming skills , for advertisers for marketing their products to reach and know about its

products , for marketers who want to earn by advertisements , smart social media to publicize their products.

5. Issues while developing the system :

While linking the game through online there may be many bugs which may attack the system and hence may damage system or may slow down the process. We need to frequently check requirements that needed to be satisfied, and

- Integration and interface issues.
- Multiple and complex user level requirements.
- For those who are unaware of computer system we need to provide login details at very basic level and also provide help box.
- For kids we need to program it as funny as it's possible introducing levels.
- Globalization causing high competition.
- Internal sourcing or outsourcing.
- We need to interact with people for understanding the needs required by them which are contradictory and should be discussed and program them at optimistic level for all.
- Sufficient software requiring specific expertise.
- Difficulty attracting and retaining applicable talent.
- Multiple software bug testing & resolution iterations.

6. Plans for developing till end of the project :

6.1. Analysis:

- ♦ Primarily,we need to understand the user requirements and interpret them.
- ♦ We need to write the functional and non-functional requirements and also the hardware and software required to make it.
- ♦ We need to estimate the budget and also time expenditure.
- ♦ We need to attend people and also need to write our program editable so that at any point of time we may further change our program for adding further defined requirements and also design it for future feasibility.

6.2 Design:

- ♦ Design is planned in object oriented model.
- ♦ Once a game is started, the user can only play the game or quit the game. He cannot open a new game.

- ♦ If the game is a win or a lose, the server should immediately display the winner, that is the server should be fast enough to display the winner without allowing the other player to make a move.
- ♦ This game is implemented in client server manner. 2 players will be playing at the client side-one playing as 'X' and the other playing as 'O'. The server will check and validate whether or not the game is a win or a draw. If it is a win, the server declares which player is the winner.

6.3 Implementation :

- ♦Implementation is done at various levels.
- ♦Base level i.e after developing basic code.
- ♦ Intermediate levels: i.e after developing certain functions(necessary)
- ♦ Final level: i.e developing the whole code.

6.4 Testing :

It is done after final level implementation.Further explanation is provided below in end.
→ After testing is done, we need to find efficient methods needed for publicizing the project at various levels also basic levels too.

7. Functional requirements :

This can be done in any required language like java,c++. Here are some of the necessary functions.

Name of the function	Input	Output	Requirement
Login class	Name , age, interests, country,email or fb linking.	User account created	Further usage is made easy by logging in through these details
Liking class	His interests for locality, his wished	Connect him with particular people and	By this we can increase his liking for

	likings in game	display his likings.	the system
User defined playing	His liking to play offline or online or with other person(specified) or with bots.	Connect him with the desired one. If he/she is not online available Notify him.	This helps customer To connect with desired one's.
Join game	Runs after calling create game.	If created shows connected to the client, else	It helps them to connect.
Create game	Whether he needs to create a game with desired person or not.	If yes creates a game with id.	Helps the other client to join game with id provided.
playturn	Inputs are defined by users while playing by clicking specifying buttons.	Helps them to play	Make moves in games
Game ended	Input is winning or tie .	Game ends.	Helps them to come out of game if ended winning or tie.
Display grid	After join game.	Displays a 3*3 grid.	It is on this grid the players take turns marking spaces as X and O..

Altering mode	After game ends,if user starts another game.	Alters X and O for players i.e complement of previous play.	To ensure that a player don't play twice, and for 1st chance.
Display score	After clicking exit game.	Displays score.	This ensures game winning with scores.
Play again	By clicking play again button	Creates another game without id.	If players want to play again.

8. Non-Functional Requirements :

8.1 Security Requirements:

- We need to take care of safety measures like of data can be hacked or erased by default by system defaults.
- We need address on security issues, sometimes our data can be hacked.

8.2 Availability:

- If the internet service gets disrupted while sending information to the server, the information can be send again for verification.

8.3 Recoverability :

- Whenever a player leaves in the middle of the game,then we should show the information on the screen for the player who is still online and we should exit the game.So,the user can play another game without bothering about this.

8.4 Performance Requirements :

1. **Responsiveness:** The game application must respond quickly to the players and changing environment.

8.5 Software Quality Attributes :

1. **Responsiveness:** The game application must respond quickly to the players and changing environment.
2. **Usability:** The game application must be easy to use so that even the new players can adapt easily.
3. **Reliability:** Ability to retrieve the scores even after the game application crashes.
4. **Portability:** The application must be portable so that it can run on any platform.
5. **Robustness:** The application must be robust to allow the scope of making mistakes by the players.

8.6 Business rules:

► Business rules must be put in mind before proceeding like budget facility and also like branding ,popularity.

9. Tentative hardware and software environment :

9.1 Hardware :

It is a online media but also provided offline for users specification. For online playing mode one requires data connection or a network.

- …> Music system is interfaced to our game application.
- …> Processor: 1.2 GHz and more.
- …> RAM: 512MB and more .
- …> Storage: 100 MB.

9.2 Software :

- …> Operating System: Windows 7,8,10 etc.
- …> Graphics: 2gb or more.
- …> Drivers: Will be provided in website's help manual.

10. Publicizing the product :

- We need to publicize this product at every levels like education,employment ,business ,market fields.
- Publicizing through internet multimedia and also through television advertisements.
- It's completely free of cost only required is a good internet connection.

10.1 Chat Session :

We have chat session while playing and we can include funny emojis so that it can ignite interest in them.

10.2 Providing game sounds :

The game must also provide background sounds. The users also has the facility to mute the sounds whenever he/she pleases.

10.3 Providing Instructions to play the Game :

For players who are new to the game, instructions are given, so that they too can play. Hints are to be given whenever the player requests hints

10.4 Dividing game into levels :

Also we can divide the game into levels based on the time taken per move and points(coins) are given according to their performance in game so that they can show more interest. We can also keep an option random battle.

10.5 Providing a social media exposure :

Also we can give an opportunity to share their score in Facebook and other social platforms.

10.6 Ease of login :

Users can login directly through gmail or facebook if they are hesitant to fill again the signup form. Thus, they can access game just by a single click.

10.7 Features :

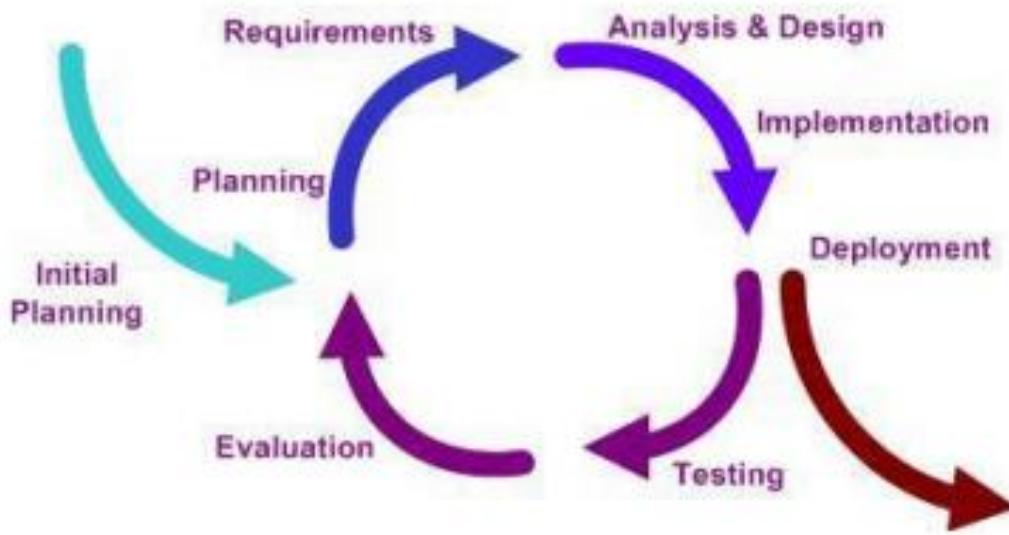
The game has multiplayer,single player and has different themes in background and has random battle.

****Extra points we gathered about the planning and testing of project****

11. Development Process :

11.1 Main Aspects :

We will plan the project over a period of 60 days and divide it into four iterations. We will plan the first iteration for analysis, second iteration for game design, third iteration for coding and the final iteration for testing of the product.



11.2 Analysis :

In the **first iteration**, we will focus on Project analysis determined as the first planned milestone of the project. Analysis is essential for starting of upcoming milestones and delivering a finished project on time. Successful completion of a project is heavily dependent on effective analysis.

11.3 Game Design :

The **second iteration** will be started by brainstorming among us on what the game would be. Each group member denotes the attributes or properties of the game that one dreamed to implement. We will gather suggestions together and choose the ones that was

possible to be implemented within a 60 days project time. As soon as the game concept become clear, we will make some early decisions on basic requirements of the project in order to more easily reach the development goals. Game design document was meant to be a living document. In other words, throughout the production process the document should be updated, if needed.

11.4 Coding :

In the **third iteration** coding in Java using Android studio was under way. In this iteration, we needed to achieve four milestones each was dependent on the previous one.

11.5 Testing :

Last iteration was planned for testing and finalizing the product. The testing process is an iterative process. We will perform the testing process in four iterations. The successful testing process of software requires a good plan. Therefore, after the requirements of the project are confirmed, the future testing of the system and the code can be planned. The test plan provided information on how and when the testing will be executed. In the second iteration, test cases can be designed for the planned tests. In iteration three, the designed test cases will be executed alongside the module testing and usability testing. During the last iteration, according to the result of the tests, the test reports will be documented properly and the bugs will be reported after the testing is completed.

12. Testing :

→ Testing is a process of executing a program with the intent of finding an error. Testing is a crucial element of software quality assurance and presents ultimate review of specification, design and coding. System Testing is an important phase. Testing represents an interesting anomaly for the software. A good test case is one that has a high probability of finding an as undiscovered error.

→ Test Levels The test approach is divided into three main phases:

Module testing, integration testing and system testing. In addition, the system testing includes two sub-phases: functional and usability testing. These planned tests are explained briefly below

(a) Module testing :

It will perform during coding by using debug messages to check that the written code produces wanted results. An important requirement is that the code will compile with zero bugs.

(b) Integration testing :

It will perform after finish module testing in order to validate if each module can work fine with each other. Integration Test proves that system works as integrated unit when all the fixes are complete.

(c) System testing :

It includes two phases functional testing and usability testing. These will perform after the product reaches its final version. During functional test phase, the tester will test if the product meets the game requirements. The tester tests the requirements using the use cases listed below in Test Cases section. The usability test will perform to understand how easy it is to learn to play the game. Any person out of the team members will perform this test by playing the game.

13. Use Case Diagram :

@startuml

:Main server: as Admin

left to right direction

player1 -> (login/signup)

player2_randomplayer ->(login/signup)

(login/signup) --->(verification)

(verification) --->(login/signup)

(verification) -->(mode)

(mode)-->(online)

(mode)-->(offline)

(online)-->(selected player/random battle)

(offline)-->(play with system)

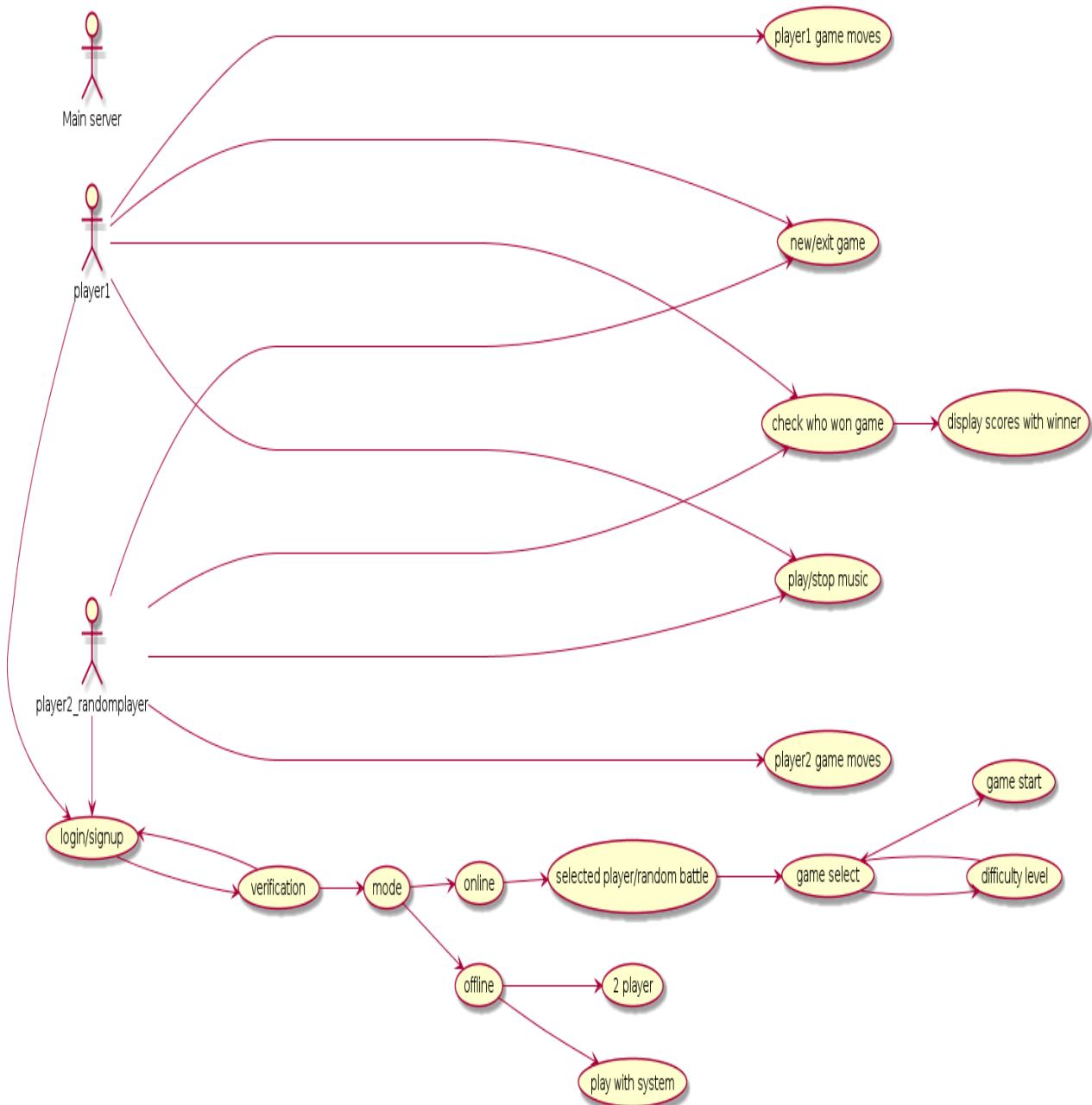
(offline)-->(2 player)

(selected player/random battle)-->(game select)

(game select)-->(difficulty level)

(difficulty level)-->(game select)
(game select)-->(game start)
player1 ----->(player1 game moves)
player2_randomplayer ----->(player2 game moves)
player1----->(check who won game)
player2_randomplayer----->(check who won game)
player1----->(play/stop music)
player2_randomplayer----->(play/stop music)
player1----->(new/exit game)
player2_randomplayer----->(new/exit game)
(check who won game)-->(display scores with winner)

@enduml



Software Requirements Specification

Document Browser(DocBros)

Version 1.0

Prepared by

**Siddhant Agarwal
(17CS30035)**

8100199177 - agarwalsiddhant10@gmail.com

**Deepank Agrawal
(17CS30011)**

7384429654 - deepank.361998@gmail.com

Indian Institute of Technology, Kharagpur

Table of Contents

Introduction	4
Purpose	4
Prospective Users	4
Need	4
Challenges	4
Future Work	4
Overall Description	5
Product Functions	5
Operating Environment	5
Design and Implementation Constraints	5
System Features	5
Creating an Account	5
Search for given set of keywords	6
Ranking the webpage	6
Annotate a personal document	6
Comment on a personal document	7
Provide extra storage and manage the storage	8
Annotate and comment on a global document	8
Adding a document to the browser	9
Other Nonfunctional Requirements	9
Safety Requirements	9
Security Requirements	9
External Interface Requirements	10
User Interface Requirements	10
Hardware Requirements	10
Software Requirements	10
Business policy	10
Publicizing the product	10
Profit	10

Revision History

Name	Date	Reason For Changes	Version
DocBros	08.02.2019	Initial abstract	1.0

1. Introduction

1.1 Purpose

A customizable document browser can be of a great utility today. A person using this system can not only browse for the documents based on a certain set of keywords, but he can rank, based on his likes and dislikes, and annotate the documents as well. The purpose of this system is basically to enable user-friendly browsing experience.

1.2 Prospective Users

Anybody who is willing to browse for documents and desire a customized experience are welcome to use the system.

1.3 Need

Almost everybody, may he be a student or an industrialists, browse for documents online. The problem is that what a student desires from a particular query is very different from what the industrialists wants. They want the search results to be ranked according to their needs. Today search engines like google try to address this problem using sophisticated information retrieval techniques that rank the results based on the their popularity but this does not guarantee much of customization. Also these browsers do not allow personalized annotations and comments. In such circumstances, a document browser that allows these features can make browsing very easy and provide a personalised browsing experience.

1.4 Challenges

The challenges that this system faces is firstly to create a browsing experience that can match that of the contemporary browsers. Also providing personalized experience to everybody is also a great difficulty. Different users will have different different ranking suggestions so providing the results as per the ranking provided by the user will be a challenging task. Also storing the personal annotations and comments as well as providing the users the freedom to comment on the global document can lead to several problems.

1.5 Future Work

Currently keywords provided by the user are simple text but we are planning to extend it to verbal as well as pictorial inputs. Also we are planning to incorporate learning methods to provide suggestions in the searches. Also the use of Neural IR can lead to better search results and more customizable experiences.

2. Overall Description

2.1 Product Functions

This project will be a Graphical User Interface at the client's end. Every user will have an account in the system. The user will be asked to enter a set of keywords for his document search. A list of documents will appear. The user will be allowed to rank the documents to modify the search results. The user is also allowed to annotate and put comments on the document, both on a personal copy, as well as in the global copy (for special members only). For personalized annotations and comments, some free storage will be provided by the server. The space can be increased by paying an additional fee. The user will also be able to download and print documents from the browser.

2.2 Operating Environment

This will be a web based system. It will have a server that will actually perform all the information retrieval and will store the customizable details of the user., The user end will simply be a graphical interface.

2.3 Design and Implementation Constraints

Currently, the documents are limited to pdfs only. Later, it will be expanded for more document types. Also the input from the user will be text based.

3. System Features

3.1 Creating an Account

3.1.1 Description

This feature is the first step for any user to browse for documents. They will require to create an account. Also special membership can be taken anytime by an additional payment. Special membership will allow users to handle the global documents as well.

3.1.2 Functional Requirements

Req 1: Ask for user details.

Req 2: Ask if the user wants to pay for extra storage.

Req 3: Allocate space in the server memory.

Req 4: If the user wants to pay, an extension to redirect to the payment gateway.

Req 5: A function to provide a secure username (unique) and password to the user.

Req 6: A function to provide an extension to keep the user logged in.

Req 7: Ask for special membership and if yes, redirect to payment gateway.

3.2 Search for given set of keywords

3.2.1 Description

This feature is the basic feature of any browsing system.

3.2.2 Stimulus/Response Sequences

The user enters a set of keywords to search for a given document. They keyword can be text or audio.

3.2.3 Functional Requirements

Req 1: Accept the keywords from the user.

Req 2: Use speech-to-text to convert speech to text(if applicable).

Req 3: Tokenization and Stemming.

Req 4: Getting a list of documents from the processed data.

Req 5: Checking if the user has defined a particular ranking order for the documents, if yes, get that order.

Req 6: Checking if user had annotated any document, if yes, retrieve the document from the personal space, if not, retrieve from the global space.

Req 7: Ask if the user wants to retrieve the global doc, if yes, than retrieve from global space.

3.3 Ranking the documents

3.3.1 Description

This feature is unique to our software system. It is one of the major feature that makes our browser so customizable. Each user can rank the documents according to their priority.

3.3.2 Functional Requirements

Req 1: Asking the user about the priority of the document, on a scale of 10.

Req 2: If two documents with same scale is obtained, then the frequency of visiting the document will decide the order.

Req 3: Given a list of documents, and the account number, generate an ordered list of documents based on the sorting order discussed.

3.4 Annotate a personal document

3.4.1 Description

Each user will be able to annotate a document. A personal copy will be stored in the provided personal space of the user.

3.4.2 Stimulus/Response Sequences

A button exist called annotate personally. The user clicks on the button. He can highlight the portions necessary. He can erase or edit the annotation as well by double clicking on the annotated area and suitably selecting from the dialog box.

3.4.3 Functional Requirements

- Req 1: A button that activates an event. An icon appears and the user hovers the icon over the region he has to highlight.
- Req 2: Save the copy of annotated document in the personal space provided.
- Req 3: If enough space is not left, intimate the user.
- Req 4: An event initiated by double clicking the annotated area. This event will invoke a dialog box to ask the user if he wants to erase or edit the annotation.
- Req 5: Button event that removes that annotation.
- Req 6: Button event that enables to edit that annotation.

3.5 Comment on a personal document

3.5.1 Description

Each user will be able to comment on specific lines on the document. A personal copy will be stored in the provided personal space of the user.

3.5.2 Stimulus/Response Sequences

A button exist called comment personally. The user clicks on the button. He can then select specific portion to add his comment in the dialog box that appears. The commented portion remains highlighted. Also he can erase and edit the comment later.

3.5.3 Functional Requirements

- Req 1: A button that activates an event. An icon appears and the user double clicks on the region where he has to comment.
- Req 2: A dialog box appears to add the comment and once saved, the commented area appears as highlighted.
- Req 3: Save the copy of commented document in the personal space provided.
- Req 4: If enough space is not left, intimate the user.
- Req 5: An event initiated by double clicking the commented area. This event will invoke a dialog box to ask the user if he wants to erase or edit the comment.
- Req 6: Button event that removes that comment.
- Req 7: Button event that enables to edit that comment.

3.6 Provide extra storage and manage the storage

3.6.1 Description

Each user can increase the amount of the personal storage provided by paying a small fee. In turn he can remove saved documents from the storage.

3.6.2 Stimulus/Response Sequences

The user will always be shown how much space he has. He can manage the storage as well as increase it. Increasing the storage is chargeable.

3.6.3 Functional Requirements

- Req 1: A menu that can take the user to the account settings page that has a storage management column.
- Req 2: Display the used and free space amount.
- Req 3: Button to direct him to manage the storage. Clicking on a button leads to the list all the documents that he had saved (annotated or commented).
- Req 4: Options to delete the saved documents.
- Req 5: An option to increase the size of the storage provided. The user has to enter the amount of the additional space required. The amount he has to pay will be displayed.
- Req 6: Link to payment gateway.
- Req 7: Upon successful payment function to allocate the required amount of space.

3.7 Annotate and comment on a global document

3.7.1 Description

Each user will be allowed to annotate and comment on the global document. This will be enabled if the user wants to highlight something about that document to the entire community. Special membership required for that which is chargeable. Also any user can report an illegal annotation or comment made by any user. But such reports can be verified by another member only. If the report is verified to be true, the user who made such a comment will be debarred from the special membership. He will have to pay again to take the membership again. However, the person can reply with proper justification. If the justification is accepted, then the person will be exempted. Each member can verify the reports and the justifications.

3.7.2 Stimulus/Response Sequences

Every user will have the freedom to report against any user who has made an annotation or comment on the global document using a button. Each member can view and verify the reports and the justifications. If a report is filed against any user, he will be immediately notified through an email.

3.7.3 Functional Requirements

- Req 1: A button will exist for the user with special member to annotate or comment on the global document.
- The annotation and comments have same requirements as already mentioned.
- Req 3: Save the copy of annotated and commented document in the global space.
- Req 4: If any report is filed against any user then immediate mail to be sent.
- Req 5: Every user who is reported against can give a justification.
- Req 6: Every user with special membership can verify the reports and the justifications.
- Req 6: Any user can file a report.

Req 7: If reports against a user is verified true and the justifications are verified false then he will be debarred from membership.

3.8 Adding a document to the browser

3.6.1 Description

Each user can add documents to the browser. The document will be added to his personal space as well as the global space.

3.6.2 Stimulus/Response Sequences

This feature will be enabled by going to the account settings menu. On selecting this option, a dialog box will appear where the user can select any document from his local system.

3.6.3 Functional Requirements

Req 1: A menu that can take the user to the account settings page that has an option for adding a document. Clicking on that menu allows the user to add a document from the local system to the browser.

Req 2: Adding the document in the personal as well as the global space.

4. Other Nonfunctional Requirements

4.1 Safety Requirements

There is no risk of any threat that can be issued by the system. Since it is a web based application, there will definitely be chances of internet based threats. These need to be minimized as much as possible. A virus scanner will scan all the documents before a user can download it. Also if a person is adding a document into the system, then also the document will be scanned for viruses.

4.2 Security Requirements

All the documents stored for a particular user will be encrypted by using the standard encryption algorithm.

5. External Interface Requirements

5.1 User Interface Requirements

There will exist an interactive Graphical User Interface. Text boxes and button events will be created where ever possible. All errors to be displayed using dialog boxes. All the main menu and settings options such as account settings, storage management etc will be always available to

the user. Most of the button events will have additional selections to be made, this will be done through an interactive dialog box.

5.2 Hardware Requirements

The server will have a very large memory (20 TB). It should be connected to the internet via high speed cables. It will have decent processing unit (6 cores at 3 GHz average clock speed) and a good graphical processing unit to enable efficient and fast computation of a few modules. The user side need to be at all sophisticated. Any device with an internet connectivity and pdf viewer will be suitable for using the system.

5.3 Software Requirements

The software requirements are not much for the user. The user should have an access to an internet browser and a pdf viewer.

6. Business policy

6.1 Publicizing the product

The basic use of this product is absolutely free of cost. One needs to pay for only the additional features. A collaboration with microsoft or google can be made that will enable them to distribute copies of the product along with their Internet Browser.

6.2 Profit

First of all the cost the users will have to pay for the extra features will be low and will aim to only balance the hardware and software costs beared by this enterprise. Moreover, as already mentioned, google and microsoft will be interested to collaborate with us. We will help them in data mining and provide data for their research. Which type of documents are set to higher priority by a particular class of users can serve as a very valuable data for them. Also this data can be forwarded to several clients including advertising agencies. This data will be made very cheap for those who will be willing to collaborate with us. Moreover a new forum for research and development will be set up for the giants.

PROJECT – 27

Course Management System

- M. Yeshwanth
17CS10027
T. Sindhu
17CS10055

- 1) Course management system is a platform where instructors can put up their content, students can register to courses available and learn from them. The courses can be free or paid as per the instructor. It also facilitates tests to be

conducted by the instructors, contains grades of students, class rosters.

This software contains an interface in which instructors apply for putting up their courses and learners apply for registering to these courses. The registration for the instructors is approved by an administrator. The registrations for individual courses are approved by their respective instructors. The administrator will receive a roster of the instructors while the courses offered by an instructor are encapsulated within the instructor. The instructor receives a roster of the students enrolled, those whose registrations are pending approval. However, he doesn't get information about the other courses enrolled by the student.

The users of this system will be instructors, learners and administrators. These three classes of users need not be exclusive.

The main issues to be tackled while developing this software are related to the intricate data flow between the users. This has to be taken care of.

3) Functional Requirements of all the users:

- ⑩ create Account : A user can create his/her account. All the users will get an id as the output.
- ⑩ edit Profile : The users can edit their profile. Input is the user id.
- ⑩ forgot Password : The users can use this to reset their password in case they have forgotten.
- ⑩ log In : To log into one's account.

Functional requirements of Instructor:

- ⑩ register Course : Request by an Instructor to register for offering a course.
The instructor is the input. Output is the id corresponding to this registration request.
- ⑩ upload Content : Instructor uploads content to his course. Course id and the files to be uploaded are the inputs.
- ⑩ deregister Course : Request by an Instructor to deregister his unwanted registered or registration requested course.

- ⑩ approve For Course Request : Validation of a student request for a particular course by an instructor. The registration id of the Student and course is the input. Output is a message to the student indicating successful registration.
- ⑩ see Class Roster : Instructor can see the roster of the students enrolled in a course. Course is the input and the list of students enrolled in it is the output.
- ⑩ view All Submissions : Instructor can view all the submissions made by the students of a particular course. Course id is the input.
- ⑩ reply : An instructor can reply to any post in the discussion forum.
- ⑩ set Grade : Instructor sets grade for a student. Student and course are the inputs.
- ⑩ send Message To Admin : Instructor can send message to the administrator. Input is the message.
- ⑩ certify Student : Sends a message to the administrators that a student has completed the course and certificate should be given. Student profile, grades report, certificate are inputs.
- ⑩ see Message Log : Instructor can see the messages from the administrators.
- ⑩ delete Message From Log : Deletes message seen by the instructor and necessary action is taken.

Functional requirements of Student :

- ⑩ register For Course : Request by a student to register for a course . The student is the input. Output is the id corresponding to this registration request.
- ⑩ deregister From Course : Request by a student to deregister from a course that he was not interested in or by mistake requested for registration.

- ⑩ submit Task : Submission to a task in the course. Task and solution files are the inputs.
- ⑩ post A Question : A student can post his/her doubt in the discussion forum.
- ⑩ reply : A student can reply to any post in the discussion forum.
- ⑩ see Grade : Student can see his/her grade. Course is the input.
- ⑩ pay For Certificate : Student pays money to get a certificate. A form need to be filled by student to be submitted to the administrator. Input is the course id. Output is a message sent to the administrators.

Functional requirements of Administrator:

- ⑩ approve Course Request : Validation of an instructor request by an administrator. The registration id of the course is the input. Output is a message to the instructor indicating successful registration and the course id.
- ⑩ see Instructor Roster : Administrator can see the roster of the instructors registered. No input is given and the output is the list of all registered instructors.
- ⑩ see Message Log : Administrator can view the messages sent by several instructors and students.
- ⑩ delete Message From Log : Deletes message seen by the administrator and necessary action is taken.
- ⑩ send Message To Instructor : Sends a message to an instructor. Input is the message and the instructor id.
- ⑩ validate Request : Sends the details of the student who paid certificate fees to the corresponding instructor to validate the details given by student. Input are the details of the student, instructor id.

- ⑩ pay Instructor : Pays the instructor a share of the certificate money. It is used after fees is paid by the student.

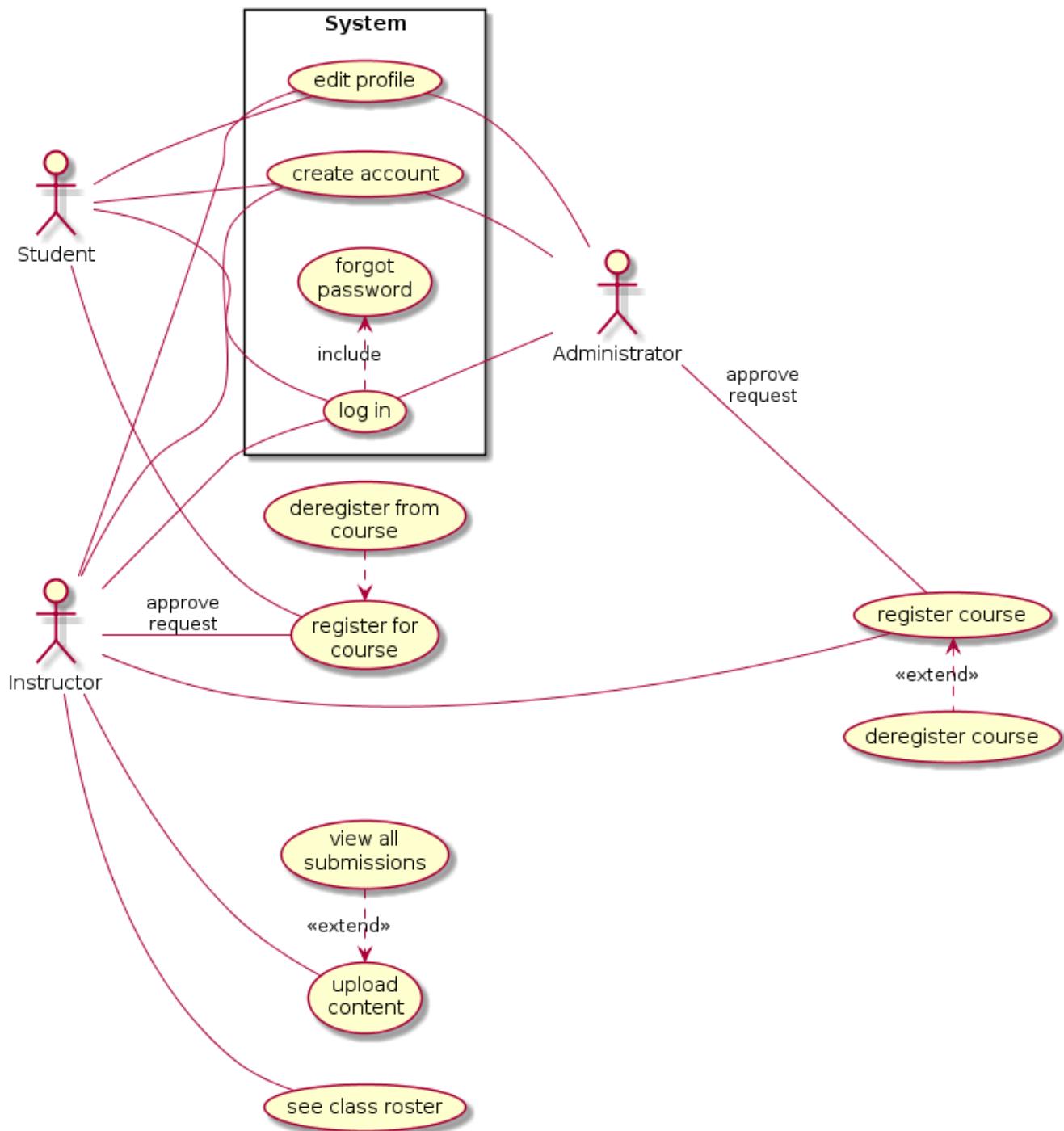
Non-functional Requirements:

- ⑩ Easy interface : because more number of users will be interested.
- ⑩ Good internet connection : The software runs on internet.
- ⑩ Response time : Average response time should be less than 5 seconds.
- ⑩ Help : help regarding how to use the website.
- ⑩ Capacity : Number of concurrent users is 1000.
- ⑩ Reliability : maximum number of failures in a year is 3.
- ⑩ Utilization of resources : Number of entries stored in the database are maximum one million. Old data is deleted in case of entry of new data.

4) Java, HTML, JavaScript, SQL, PHP are required. SQL is required for DataBase Management System. HTML, JavaScript, PHP may be used for developing websites.

5) To publicize, Homepage of the website contains the esteemed faculty we have. We allow courses to be audited free but need to pay for getting a certificate.

USE CASE DIAGRAM



Software Requirements Specification

for

WordPro Duo

Version 1.0

Prepared by

Rajat Kumar Jenamani
17CS10061
rajatkj11@gmail.com

and

Jyotisman Das
17CS10017
jeetjyotismanjeet1999@gmail.com

Table of Contents

Table of Contents	2
Revision History	2
1. Introduction	3
1.1 Purpose	
1.2 Document Conventions	
1.3 Need	
1.4 Intended Use	
1.5 Product Scope	
1.6 References	
2. Overall Description	5
2.1 Product Perspective	
2.2 Product Functions	
2.3 User Classes and Characteristics	
2.4 Operating Environment	
2.5 User Documentation	
2.6 Assumptions and Dependencies	
3. External Interface Requirements	7
3.1 User Interfaces	
3.2 Hardware Interfaces	
3.3 Software Interfaces	
3.4 Communications Interfaces	
4. System Features and Functional Requirements	10
4.1 Autocomplete Bar	
4.2 Control Keys	
4.3 Main Input Keys and Navigation Keys	
5. Other Nonfunctional Requirements	12
5.1 Performance Requirements	
5.2 Safety Requirements	
5.3 Security Requirements	
5.4 Software Quality Attributes	
5.5 Business Rules	
6. Other Requirements	13
7. Project Plan	13
8. Publicity	14

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

WordPro Duo strives to provide the user with a **seamless smart virtual keyboard experience across interfaces and devices**. It **supports various languages** and allows people with **mobility impairments** to type data by using a pointing device. WordPro Duo also **protects the user from being spied** from sources which record keystrokes.

1.1 Purpose

This document is the **Software Requirement Specification** for WordPro Duo. The SRS document aims to **explain the software** in an easy manner, the basic idea behind the smart virtual keyboard and how the developers aim to achieve their goals. It also aims to **introduce to the users the main features of WordPro Duo**. The document **contains overall description, functional requirements and non functional requirements**.

1.2 Document Conventions

Standard document conventions have been used throughout the document.

1.3 Need

WordPro Duo is a much needed software in today's fast paced world as it is **built in to the device and no wires or separate device with extra batteries** are required. It **saves extra weight** on a portable device and helps in **reducing the risk of keystroke logging** which is of great significance in this era of privacy.

Users often find it **difficult to input operations into a calculator on a standard keyboard**. WordPro Duo provides the user with an option to switch between **different modes available such as math mode and text mode**. Each of these modes has **features and keyboard layout specific to the work in context** which makes working with WordPro Duo a seamless experience.

The keyboard also provides an **essential role for people with a wide variety of motor disabilities**. People who have conditions that affect dexterity, illnesses that cause pain and fatigue and limited capacity for body movement, might find it **difficult or impossible to use a mouse because it requires too much coordination and precision**. A **keyboard that works with a screen pointer** might offer them a more controlled means of input that does not require extensive movement.

1.4 Intended Use

WordPro Duo is primarily aimed to act as an **on-screen input method** in computers and other **devices with no physical keyboard**, where there is no room for one, such as a **pocket**

computer, tablet computer or touchscreen-equipped mobile phone. Text is commonly inputted either by tapping a virtual keyboard or by using a screen pointer.

1.5 Product Scope

WordPro Duo is a **free, multilingual and customisable smart virtual keyboard**. It lets you type directly in your local language script in an easy and consistent manner, no matter where you are or what computer you're using. Some common uses of virtual keyboards include:

- Allowing a person to type in their **own language on foreign lands**- such as when traveling abroad or living in another country.
- Enabling a more accessible typing experience by allowing **typing by on-screen clicks**.
- Contains different **modes like calculator mode and text mode**.
- Allows people with **mobility impairments** to type data by using a pointing device.

1.6 References

1. https://en.wikipedia.org/wiki/Virtual_keyboard
2. <https://github.com/hodgef/simple-keyboard>
3. <https://whatis.techtarget.com/definition/virtual-keyboard>

2. Overall Description

2.1 Product Perspective

WordPro Duo is a versatile smart virtual keyboard that provides the user with a seamless typing experience.

2.2 Product Functions

Key features:

1. Support of **multiple regional languages**.
2. Has various modes like **text mode and calculator mode**.
3. **Word autocomplete** to make typing faster and more accurate than ever
4. Fully **customizable** look and behavior
5. Allows people with **mobility impairments** to type data by using a pointing device.

2.3 User Classes and Characteristics

WordPro Duo is a versatile virtual keyboard with an inbuilt calculator interface. The software can be used by anyone as it is as easy as using a smartphone. The pointing device needs to be used normally and choose the characters from a wide variety of encodings. The interface is very user friendly incorporating two modes - the text mode and the calculator mode.

Apart from **mainstream public**, WordPro duo can also be used by the following user classes-

- a. **People who have mobility impairments** find it difficult or impossible to use a mouse because it requires too much coordination and precision. A keyboard that works with a screen pointer might offer them a more controlled means of input that does not require extensive movement.
- b. **People who are not proficient in English** and thus cannot use the normally available Keyboard with English keys. WordPro duo supports various regional languages and thus is of great use to this user class.
- c. **Mathematicians who frequently need to write text involving mathematical symbols** but cannot do so on a normal keyboard as keys for many symbols aren't present. WordPro Duo can switch from the default text mode to calculator mode at just the press of a key and thus allow typing of symbols not usually found on normal keyboards.

2.4 Operating Environment

WordPro Duo - Version 1.0 is presently compatible with **Ubuntu 16.04 and above**. Efforts are being made to make it compatible with Mac OS and Windows.

2.5 User Documentation

WordPro Duo comes with a user documentation that will guide the user through the functions and various features available.

2.6 Assumptions and Dependencies

1. It is assumed that the user have provided the software with **proper privileges and not blocked** it from starting automatically.
2. The application **stays active in the background** consuming a very small amount of memory which is assumed to be negligible as compared to general memory constraint required to run a PC.
3. It is assumed that the user has an **access to at least one pointing device** (mouse, stylus, light-pen) otherwise the software won't be accessible.
4. To get suggestions in parallel to typing it is necessary to create a database, i.e. a word pool that consists of the words mostly used by the user. It is assumed the software has the **access to the disk** to store this data pool.
5. It is assumed that the user has a **proper knowledge regarding the input format of the interface**. For example, if the interface demands an english input but the user tries to feed in using some other language then the software might not input anything or might even crash.

3. External Interface Requirements

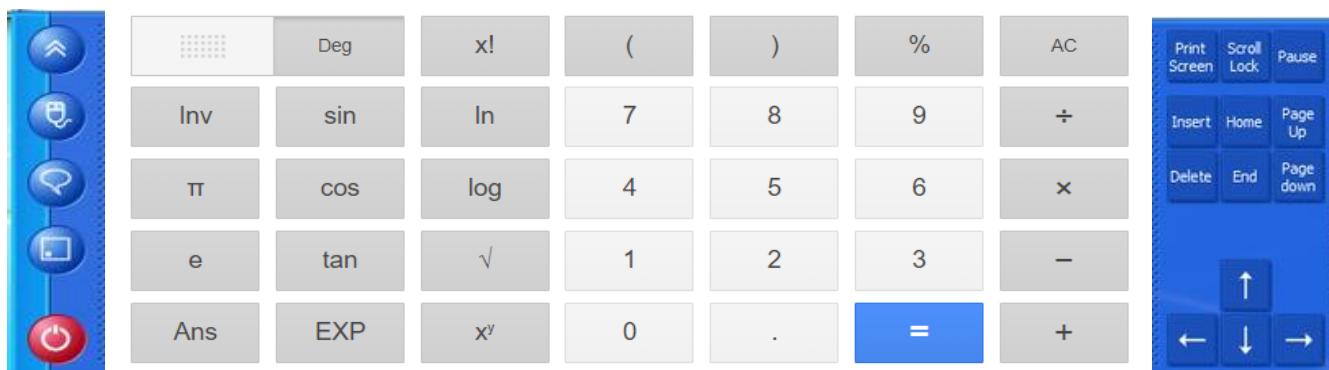
3.1 User Interfaces

The user interface will look like the following images:

Text Mode -

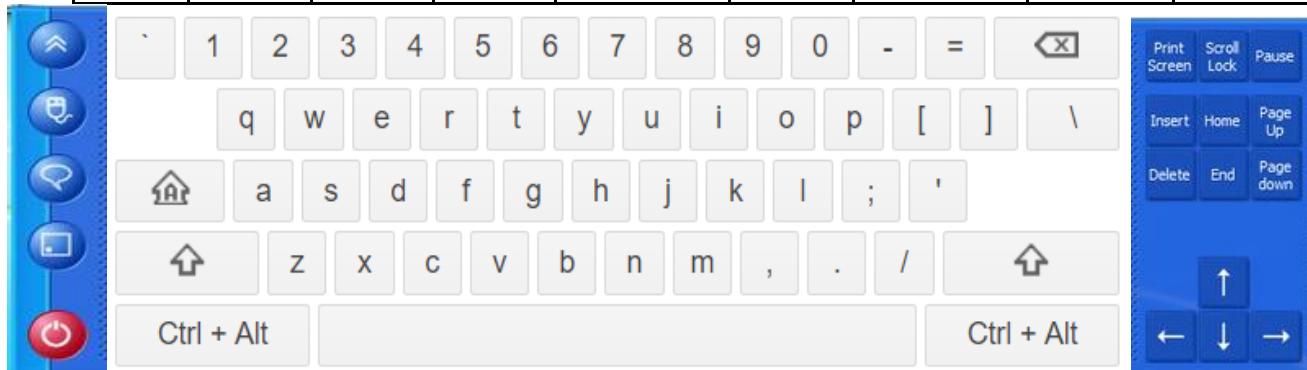


Calculator Mode -



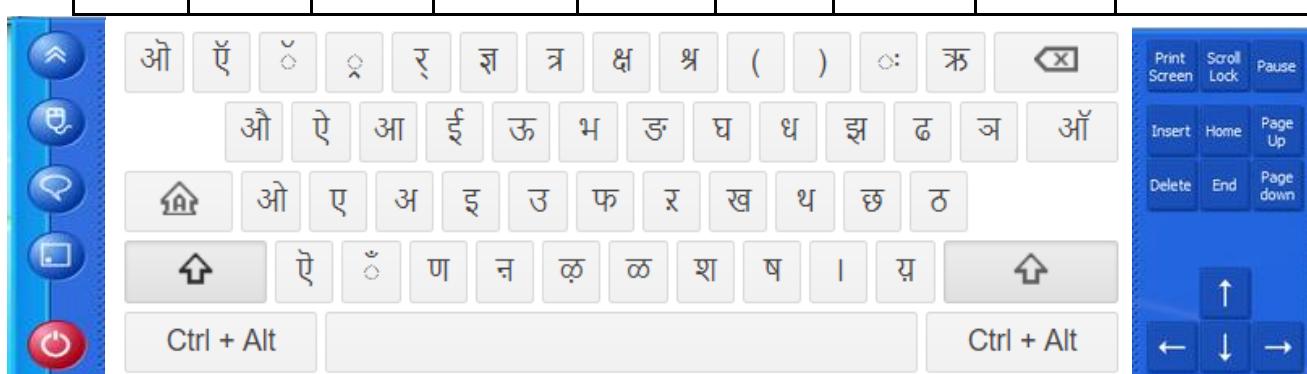
Language: English (Text Mode) -

who	why	when	what	whatever	while	whether	where	which
------------	------------	-------------	-------------	-----------------	--------------	----------------	--------------	--------------



Language: Hindi (Text Mode) -

अगला	अग	अकेला	अनजान	अजनबि	अनेक	अलग	औरत	आम
-------------	-----------	--------------	--------------	--------------	-------------	------------	------------	-----------



3.2 Hardware Interfaces

1. A proper display
2. A pointing device (mouse, stylus, light-pen)
3. Data storage (lesser than 1GB) access to create the word pool

3.3 Software Interfaces

1. An OS based on **Ubuntu 16.04 and above** is required for the software.
2. The **Java APIs Swing and JavaFX** were used to create the beautiful user friendly GUI of WordPro Duo.
3. **Swing event handling** was incorporated.
4. Swing was used as it has a greater number of packages than AWT.

3.4 Communications Interfaces

The software is **standalone and does not require internet** for its functioning although the user has an option to create an online WordPro Duo account and connect it to the software. The account will take backups of the words frequently used by the user and will come handy if the user handles multiple machines provided internet access is available.

4. System Features and Functional Requirements

This lists all of the features and functional requirements of WordPro Duo with proper explanations.

4.1 Autocomplete Bar

On the top of the keyboard is the text suggestion bar.

4.1.1 Description and Priority

Autocomplete is a feature in which an application predicts the rest of a word a user is typing.

Priority : Medium

4.1.2 Stimulus/Response Sequences

Autocomplete or word completion works so that when the writer writes the first letter or letters of a word, the program predicts one or more possible words as choices from a database of prospective words. If the word he intends to write is included in the list he can select it, for example by using the number keys. If the word that the user wants is not predicted, the writer must enter the next letter of the word. At this time, the word choice(s) is altered to suite the new word. When the word that the user wants appears it is selected, and the word is inserted into the text.

4.1.3 Requirements

A wordpool database that initially contains a fixed number of words. Any new words input by the user and not present in the wordpool are added it.

4.2 Control Keys

The control keys, which are present on the **left side of the keyboard**, can be used to alter the settings of the keyboard.

4.2.1. Description and Priority

The control keys can be used to use the various features of WordPro Duo. The following keys are available -

1. **Customize Key:** Allows the user to customize the keyboard layout and create his/her own keyboard layouts.
2. **Input Key:** Allows the user to switch between a mouse and a screen pointer as input options
3. **Language Key:** Allows the user to change the language of the keyboard. WordPro Duo version 1.0 will support English and Hindi.
4. **Zoomable Key:** Allows the user to control the size of the keyboard
5. **Mode Key:** Allows the user to switch between two modes:
 - a. Text Mode
 - b. Calculator Mode

Priority : High

4.2.2. Stimulus/Response Sequences

1. **Customize Key:** Clicking on the customize key will take the user to a new window which allows the user to select the appearance of the the keyboard liked.
2. **Input Key:** The default input method is mouse. Clicking on the input key will switch the method of input to a screen pointer and another click will switch it back to mouse.
3. **Language Key:** The default language is english. Clicking on the input key open a list of languages that the user can choose from.
4. **Zoomable Key:** Clicking on this button will open a slider that the user can use to control the size of the keyboard. After selecting the size the user can click on Apply which will change the size of the keyboard to the desired size.
5. **Mode Key:** The default mode is Text Mode. Clicking on the mode key will switch the mode to Calculator Mode and another click will switch it back to Text Mode.

4.2.3. Requirements

A list of languages that the keyboard supports

4.3 Main Input Keys and Navigation Keys

The **main input keys** are present on the **centre of the virtual keyboard** and the **function keys** are are present on the **right side of the virtual keyboard**.

4.3.1 Description and Priority

The main input keys are the qwerty keys along with the numbers and the special characters. Navigation keys, as the name suggests are used to navigate the cursor through the interface.

Priority : High

4.3.2 Stimulus/Response Sequences

The main input keys are the ones that feed-in the characters in the interface when pointed or clicked on one of them and the navigation keys execute the designated function.

4.3.3 Requirements

A pointing/clicking device which is used to select the characters that are to be fed-in.

5. Nonfunctional Requirements

5.1 Performance Requirements

5.1.1 Response Time

The software has a very low response time (lesser than 0.05 s) and inputs instantaneously at the interface. The calculator mode has a slightly higher response time (about 0.1 s) but still incredibly fast.

5.1.2. Work Load

Since the software is meant to serve a single user at a time, there is not any significant workload variation pressure and as such the keyboard functions seamlessly.

5.1.3. Scalability

WordPro Duo is a standalone single user based software hence scalability isn't an issue at all. The keyboard is fired up in its default mode as soon as the user hits the pointer at an inputtable interface.

5.1.4. Platform Considerations

Presently the software is supported on Ubuntu 16.04 and above. Windows version shall be out soon.

5.2 Safety Requirements

1. The safety part of the system will be based on the facilities provided by the OS .
2. Environmental Conditions
3. Operating Regimes (start up, shut down)
4. Failure modes and protection requirements

5.3 Security Requirements

The security requirements of the software are handled by the OS used.

5.4 Software Quality Attributes

The Quality of the system is reflected from its friendly user interface with minimized resource utilization.

The following software attributes represents quality attributes:

1. Secured
2. Maintainability
3. Fast
4. Memory efficient
5. Reliable

5.5 Business Rules

1. The software is absolutely a freeware which the user can use in any of his/her work whether be it daily life or office presentations.
2. If the GUI is not triggered automatically, make sure to turn off any conflicting softwares which might be stopping WordPro Duo from being active in the background. The software has to be alive at all times to provide a seamless experience. If you are using any anti-virus make sure that it whitelists WordPro Duo.
3. The software creates a database of the words that are being used by the user frequently. The pooled database can be shared and viewed by the user by accessing the feature provided by us at our website. The user must sign up for an account at our website using his email which is absolutely free.
4. By using our WordPro Duo account the user can even add custom words to the vocabulary.

6. Other Requirements

Recoverability

The WordPro Duo account (free) associated with the user can be used to recover the personalized word pool.

7. Project Plan

1. Week 1: (Preparing SRS)

The first week was used to carefully prepare the SRS of WordPro Duo.

2. Week 2: (Listing features)

This week consists of listing all the features that are to be incorporated.

3. Week 3: (Designing)

This week consists of designing the GUI. The GUI needs to be simple but elegant.

4. Week 4: (Hard-Coding)

This week consists of hard-coding i.e. integrating all of the features with the design established.

5. Week 5: (Testing and Debugging)

The final week consisted of thorough testing of all the features and debugging the errors. Additional features if found useful shall be incorporated as well.

8. Publicity

WordPro Duo is a **free, multilingual and customisable smart virtual keyboard**. It lets you type directly in your local language script in an easy and consistent manner, no matter where you are or what computer you're using. Some common uses of virtual keyboards include:

- Allowing a person to type in their **own language on foreign lands**- such as when traveling abroad or living in another country.
- Enabling a more accessible typing experience by allowing **typing by on-screen clicks**.
- Contains different **modes like calculator mode and text mode**.
- Allows people with **mobility impairments** to type data by using a pointing device.