

A REPORT

ON

EXAMINATION HALL TICKET VERIFICATION AND AUTHENTICATION

BY

Adarsh Anand	2017B2A70090G
Krishna Datta	2017A7PS0007G
Metlapalli Venkata Sai Karthik	2017A7PS0260H
Rajath Reghunath	2017A7PS0059G
Reshma Seby John	2017A7PS1909G
Siddhant Khandelwal	2017A7PS0127P
Suraj Sudheer Menon	2017A7PS0149G

AT

Center for Development of Imaging Technology, Trivandrum

A Practice School-I station of

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

July, 2019

A REPORT
ON
EXAMINATION HALL TICKET VERIFICATION AND AUTHENTICATION

BY

Adarsh Anand	2017B2A70090G	Computer Science, Chemistry
Krishna Datta	2017A7PS0007G	Computer Science
Metlapalli Venkata Sai Karthik	2017A7PS0260H	Computer Science
Rajath Reghunath	2017A7PS0059G	Computer Science
Reshma Seby John	2017A7PS1909G	Computer Science
Siddhant Khandelwal	2017A7PS0127P	Computer Science
Suraj Sudheer Menon	2017A7PS0149G	Computer Science

Prepared in partial fulfilment of the Practice School-I Course

AT

Center for Development of Imaging Technology, Trivandrum

A Practice School-I station of

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI

July, 2019

Acknowledgements

We would like to extend our heartfelt gratitude to Center for Development of Imaging Technology, Trivandrum (CDIT) for actively engaging with us and giving us the space to build viable solutions to real world problems.

We would like to specially thank Mr. N. Jayaraj, our Station Coordinator, whose constant presence and help has helped us through completing this project and understanding the workings of a corporate environment.

We would also like to thank Mr. Shibu Manikkoth, who has been the lead for our project, answering all our queries, and helping us understand the specific requirements of the project.

Last but not least, we would like to thank our Faculty Coordinators, Mr. Rayson K. Alex, and Mr. Deepu Dileep, for their constant support, encouragement and presence.

Finally, we would like to acknowledge the role played by BITS Pilani University and the Government of Kerala, in making this internship opportunity available to us.

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE

PILANI (RAJASTHAN)

Practice School Division

Station: Trivandrum **Center:** Center for Development of Imaging Technology, City Office

Duration: 2 months **Date of Start:** 21 May 2019

Date of Submission: July 08 2019

Title of the Project: Examination Hall Ticket Verification and Authentication

Student Details

Adarsh Anand	2017B2A70090G	Computer Science, Chemistry
Krishna Datta	2017A7PS0007G	Computer Science
Metlapalli Venkata Sai Karthik	2017A7PS0260H	Computer Science
Rajath Reghunath	2017A7PS0059G	Computer Science
Reshma Seby John	2017A7PS1909G	Computer Science
Siddhant Khandelwal	2017A7PS0127P	Computer Science
Suraj Sudheer Menon	2017A7PS0149G	Computer Science

Names and Designation of the experts

Mr. N. Jayaraj	Head, Informatics Division
Mr. Shibu	Head, Software Development

Names of PS Faculty

Mr. Rayson K. Alex

Mr. Deepu Dileep

Key Words: Examination, Hall Ticket, Verification, Biometric, Fingerprint, Authentication

Project Areas: Web Development, Biometric Authentication

Abstract:

The current problem with Hall Ticket Verification involves manually checking data entered by Exam Aspirants, which includes large overheads with respect to time, manual labour and storage space. This leaves room for Data Redundancies and the possibility of Impersonation.

The current system involves printing out hard copies of Hall Tickets for every stage of every examination and long queues at Registration Desks, and the process leads to unforeseen problems such

as limitations on the kind of examination being conducted to ensure fair testing, the time taken for an examination to be conducted, and overutilisation of valuable state resources.

The steps taken so far to solve these issues include but are not limited to, making Aadhar mandatory for Verification, including necessary details on Hall Tickets, making them unnecessarily long and cluttered.

The following solution proposed effectively deals with these issues by Digitising the Verification Process to reducing the Hall Ticket to a Registration Number that references a Database which contains all the relevant details, and collecting Biometric Data to Authenticate further stages of an examination without any extra overheads.

As a single team of seven, we will be using JQuery, Node.JS, Sequelize, HTML and Bootstrap to build a dedicated FOSS project that is scalable across platforms and is efficient in solving the listed issues by following these objectives:

- a) To make sure Data required is entered only once, and accessed from a common, secure Database designed according to current industry standards.
- b) To implement a Database that is scalable and can be used for multiple examinations and purposes.

- c) To enforce a two step Implementation of the software for enabling the constraints of an Examination and identifying its specific Utility.

Signature of Students:

Adarsh Anand

Krishna Datta

Metlapalli Venkata Sai Karthik

Rajath Reghunath

Reshma Seby John

Siddhant Khandelwal

Suraj Sudheer Menon

Signature of PS Faculty:

Rayson K. Alex

Deepu Dileep

Date:

Date:

Table of Contents

1	Cover Page.....	1
2	Title Page.....	2
3	Acknowledgements.....	3
4	Abstract Page.....	4
5	Introduction.....	10
6	Development.....	12
	6.1 Frontend.....	12
	6.2 Backend.....	14
	6.3 Database Design.....	16
7	Integration.....	20

7.1 Integration.....	20
7.2 Rollout.....	20
8 Conclusions.....	21
9 References.....	22
10 Appendix A.....	23
10.1 Introduction to Software Development Life Cycle.....	23
10.2 User Requirements Document.....	25
10.3 System Requirements Specification.....	29

Introduction

In today's world, the value of a degree from a prestigious university cannot be overstated, and the students trying to prove themselves worthy are in the millions. Competition in the fields of Post Graduation and Medicine are especially immense. In such an environment, the scope of malpractice in examinations is very high, especially in the area of Hall Ticket Verification and candidate Authentication. Issues of impersonation of candidates are one such problem area.

Currently, the measures being taken to control such occurrences are limited. The candidate is required to have a physical copy of a long, cluttered Hall Ticket containing all the details pertaining to their identity and eligibility, along with a Government issued photo identity card, at all times, and are required to present it to present the same when demanded.

This method of authentication of candidates falls short of an efficient solution for several reasons. The time taken to authenticate all the candidates appearing for a certain examination is very long, making it a very slow process. The amount of state resources and human resources being employed for the same is also an unnecessary overhead that disrupts regular functioning of the government. Also, with this large volume of candidates and limited human resources, the scope for issues of Hall Ticket manipulation and malpractice going unnoticed is high.

One way to reduce such occurrences is to take a page out of the Central Government's Aadhar program, which is to use Biometric Data, specifically Fingerprints to authenticate and verify candidates.

Employing such a solution effectively deals with all the issues listed above in an efficient and secure manner. Candidates will be required to enter their relevant details online, that will be stored on a central database on a secure server. Once these details have been entered, the candidate is simply given a single Registration Number. This means that they only have to carry this to the exam centre on the day of the exam along with a Government issued photo identity card.

At the registration desk, this Registration Number is entered into the system and all the relevant details are retrieved from the database. Once this is verified, the candidate is required to submit their identity card details, which will henceforth be stored on the system, so they do not have to carry it for further stages or iterations of the examination. After this, the candidates fingerprints, all five fingers of their right hand, are taken and stored, as well as their photograph at the time of registration. Once this process is complete, for any further iteration or stage of the examination, they simply have to provide their fingerprint and registration number to have their identity verified.

This software solution being developed along these lines has been done keeping in mind current industry practices and recent trends. The major part of the development is being done in JavaScript, with the backend employing the use of Node.JS, the database uses MySQL for design and Sequelize for implementation of queries, and finally the frontend uses HTML, Bootstrap and JQuery to design a responsive and intuitive user experience.

We have tried to make this software scalable across operating systems and devices, and hope we have built a comprehensive solution that can be used to ensure a fair testing environment for examinations.

Development

The development of the software is done based on Free and Open-Source Software (FOSS) tools using the latest stable versions. The development stage is divided into three distinct branches, Frontend, Backend and Database Design. Our team has been divided into three parts such that 3 people work on the Frontend, 3 on the Backend, and one on Database Design and all relevant Documentation.

Frontend

The Frontend consists of designing everything pertaining to the Client side of the software, which includes designing secure webpages and intuitive user interfaces that offer a seamless user experience. This also includes querying relevant information from the database and updating new information as well.

This employs the following software: HTML 5, BootStrap 4.3.1, JQuery 3.4.1, CSS 3

In this case, the Frontend is divided into four webpages with distinct functions,

a) Login Page

For authorised personnel who will receive credentials that are set by the administrator or auto generated. They will be the only ones allowed to access the software functionality for a specified time interval.

b) Constraint Selection

For a singular authority, like an examination superintendent, who will be able to select the Level of the examination, the accepted identity cards as well as examination duration for which the software functionality will be enabled.

c) Level 1

For the first time a candidate appears, their details are populated according to a database query after they enter a unique registration number, then their Biometric Fingerprint data is taken as well as a photograph, as well as valid identity card details.

d) Level 2

For subsequent appearances in the same examination track, a registration number and a fingerprint will be taken, and identity will be verified.

Backend

The backend consists of designing everything pertaining to the server, including hosting the website, the database and any other soft assets required for the website. The server acts as the point of control to connect the database to the frontend. This is done through the use of REST API endpoints which are designed to create a channel of data between the client and server. The client requests to these endpoints are handled by the server and the result is returned after processing.

The backend will be built using Node.js (11.6.0) with Express framework (2.6.9) and Sequelize (5.9.4) as the ORM for MySQL (8.0).

This project uses three main api endpoints:

1. /api/login

The login api will take in two parameters - Username and Password and validate the potential user with the adminUsers table. A token will be generated which will be used by the client at every subsequent request to the server. Depending on their permission levels, the appropriate pages are served.

2. /api/select-constraints

This api will select whether level 1 or level 2 pages are to be served. Also switches on/off the Biometric update process.

3. /api/update-biometric

The client can send POST requests on this endpoint with the recorded Biometric Details on Level-1 of the exam.

4. /api/verify-candidate

This endpoint will be used to verify the Biometric details provided by the Applicant during Level-2 and subsequent levels.

Database Design

The database design consists of efficient storage of data in such a manner that data redundancies are reduced and the required information is retrieved in an efficient manner in the least time possible. In our project, there is a central database stored on the server, but during the examination, there is a local copy on the client machine that is changed according to need and the central database is updated with the new data after the examination ends.

Passwords are stored in the Advanced Encryption Standard (AES) which enables the user to customise the salt used to encrypt their password strings. AES implements two inbuilt functions in the following format,

- a) AES_ENCRYPT(“unencryptedPasswordString”, “salt”)
- b) AES_DECRYPT(“encryptedPasswordString”, “salt”)

This database design uses MySQL 8.0, Sequelize 5.9.4

There are three tables in one database in the current schema,

- a) tbl_candidates: the table is already present, implemented by the centre.

Sl. No.	Name	Datatype	Key	Attributes
1	intCandidateID	INT(11)	-	NOT NULL
2	intUserID	INT(11)	-	NOT NULL
3	vchrCandidateName	VARCHAR(30)	-	NOT NULL
4	vchrCandidateRollNo	VARCHAR(20)	-	NOT NULL
5	vchrBarcodeNO	VARCHAR(20)	-	NOT NULL
6	vchrCandidateAddress	VARCHAR(100)	-	NOT NULL

7	vchrCandidateEmail	VARCHAR(20)	-	NOT NULL
8	vchrCandidatePhone	VARCHAR(20)	-	NOT NULL
9	dteCandidateDob	DATE	-	NOT NULL
10	intDistrict_Cd	VARCHAR(2)	-	NOT NULL
11	intAlphaCodeID	INT(11)	-	NOT NULL
12	intExamDistrictID	INT(11)	-	NOT NULL
13	intExamCenterID	INT(11)	-	NOT NULL
14	intExamID	INT(11)	-	NOT NULL
15	vchrCandidateUserName	VARCHAR(8)	-	NOT NULL
16	vchrCandidtatePassword	VARCHAR(20)	-	NOT NULL
17	vchrCandidatePhoto	VARCHAR(20)	-	NOT NULL
18	intTimeExtended	TIME	-	NOT NULL
19	vchrCandidateIp	VARCHAR(20)	-	NOT NULL
20	intUserID	INT(11)	-	NOT NULL
21	intStatus	INT(11)	-	NOT NULL, DEFAULT '1'
22	chrPH	CHAR(1)	-	NOT NULL
23	chrRes_gp	CHAR(1)	-	NOT NULL
24	chrEX	CHAR(1)	-	NOT NULL
25	chrSP	CHAR(1)	-	NOT NULL
26	cand_cd	VARCHAR(12)	-	NOT NULL
27	cand_user_cd	VARCHAR(30)	-	NOT NULL
28	reserve_grp_det	VARCHAR(60)	-	NOT NULL
29	cand_id_marks	VARCHAR(200)	-	NOT NULL
30	cand_sex	VARCHAR(1)	-	NOT NULL
31	debar_status	INT(11)	-	NOT NULL
32	debar_remarks	VARCHAR(400)	-	NOT NULL
33	debar_time	DATETIME	-	NOT NULL
34	IntPattern_ID	INT(11)	-	NOT NULL
35	vchrSeatno	VARCHAR(10)	-	NOT NULL
36	intbatch_id	INT(11)	-	NOT NULL
37	tmLastActivity	TIME	-	NOT NULL

b) postVerification_table

Sl. No.	Name	Datatype	Key	Attributes
1	rollNo	INT	Primary	NOT NULL, UNIQUE
2	newPhotoPtr	VARCHAR(2048)	-	NOT NULL
3	fingerprintOne	LONGBLOB	-	NOT NULL
4	fingerprintTwo	LONGBLOB	-	NOT NULL
5	fingerprintThree	LONGBLOB	-	NOT NULL
6	fingerprintFour	LONGBLOB	-	NOT NULL
7	thumb	LONGBLOB	-	NOT NULL
8	idType	INT	Foreign	NOT NULL, DEFAULT-1
9	idNumber	VARCHAR(20)	-	NOT NULL, UNIQUE, DEFAULT “NONE”

c) idCard_table

Sl. No.	Name	Datatype	Key	Attributes
1	idType	INT	Primary	NOT NULL, UNIQUE, AUTO INCREMENT
2	idName	VARCHAR(20)	-	NOT NULL
3	idStatus	BOOLEAN	-	DEAFULT 1

d) adminUsers_table

Sl. No.	Name	Datatype	Key	Attributes
1	userNo	INT	Primary	NOT NULL, AUTO INCREMENT
2	userName	VARCHAR(10)	-	NOT NULL
3	userPassword	CHAR(60)	-	NOT NULL
4	permissionLevel	INT	-	NOT NULL
5	userStatus	BOOLEAN	-	DEFAULT 1

Sequelize is a promise-based ORM for Node.js. ORM or Object Relation Mapping is a process of mapping between objects and relation database systems. An ORM acts like an interface between two system. ORM provide advantages for developers from basic ones like saving time and effort and rather focusing on business logic. The code is robust instead of redundant. ORM helps in managing queries for multiple tables in an effective manner. Lastly, an ORM is capable to connect with different databases.

We have used Sequelize to enable this Database to be able to interact with the Backend.

Integration

This is the final process to reach an end product that is ready to be utilised. All the different tracks will integrate their code to work as a single unit and write tests to make sure it is secure and can be implemented in the outside world.

Integration

The code that is written is all pushed to a single repository on GitHub, and when the individual modules are ready, we combine them to form the complete software application. At every stage of integrating a module, we run tests to make the software application is bug free and works according to specifications and give expected results. When modules are combined, sometimes the data travelling between modules has many errors which may not give the expected results. So integration testing is performed to find the defects or bugs in all the interfaces.

Rollout

Once these phases are done, the final software application is fully functional and ready to be rolled out for use. Since this is classified as FOSS, it will be available for free and users will be allowed to add, delete or modify the code in any manner.

Conclusions

In conclusion, this software application can be used to effectively reduce malpractice issues related to Examination Hall Tickets. There may be unknown security lapses in the current model as well, that have not been identified by this particular application, so continuous monitoring and upgrades are necessary.

This software application will reduce human resource requirements exponentially, make the time required to authenticate or validate a particular candidate negligible, and uphold the validity of each such authentication as it is done using stringent protocols.

References

1. Bootstrap team (2019 February 27). *Forms & Navbars* Retrieved from <https://getbootstrap.com/>.
2. Fingerprint Image and Minutiae Data Standard for e-Governance Applications in India (2010 November). Retrieved from <http://egovstandards.gov.in/sites/default/files/Fingerprint%20Image%20Data%20Standard%20Ver1.0.pdf>
3. MySQL Consulting and NoSQL Consulting: MySQL DBA (2006 August 21) *Storing Passwords in MySQL*. Retrieved from <http://mysqldatabaseadministration.blogspot.com/2006/08/storing-passwords-in-mysql.html>
4. Ilya Kantor (2019,February). The JavaScript Language. Retrieved from <https://javascript.info/>.
5. “Node.js v10.16.0 Documentation.” Index | Node.js v10.16.0 Documentation. Retrieved from <https://nodejs.org/dist/latest-v10.x/docs/api/>.
6. “Express 4.x API.” Express 4.x - API Reference. Retrieved from <https://expressjs.com/en/api.html>.
7. “IEEE 23026-2015 - ISO/IEC/IEEE International Standard - Systems and software engineering - Engineering and management of websites for systems, software, and services information”. Applicable IEEE standard. Retrieved from <https://standards.ieee.org/standard/23026-2015.html>

Appendix A

Introduction to Software Development Life Cycle

Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop and test high quality softwares. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

A typical Software Development Life Cycle consists of the following stages,

a) Planning and Requirement Analysis

Requirement analysis is the most important and fundamental stage in SDLC. It is performed by the senior members of the team with inputs from the customer, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility study in the economical, operational and technical areas.

b) Defining Requirements

Once the requirement analysis is done the next step is to clearly define and document the product requirements and get them approved from the customer or the market analysts. This is done

through an SRS (Software Requirement Specification) document which consists of all the product requirements to be designed and developed during the project life cycle.

c) Designing the Product Architecture

SRS is the reference for product architects to come out with the best architecture for the product to be developed. Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented.

d) Building or Developing the Product

In this stage of SDLC the actual development starts and the product is built. The programming code is generated as per SRS during this stage. Developers must follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers, etc. are used to generate the code.

e) Testing the Product

This stage refers to the testing only stage of the product where product defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.

f) Deployment in the Market and Maintenance

Once the product is tested and ready to be deployed it is released formally in the appropriate market. Any vulnerabilities or leaks are fixed and updates are constantly rolled out for a specified period of time to maintain application integrity.

As part of the SLDC, documents such as the User Requirement Document and Software Requirement Specification are listed.

User Requirement Document

Goals and Contexts

Our goal is to create a web application that digitizes and streamline the process of Hall Ticket Verification and Authentication using Biometric Data.

The current problem with Hall Ticket Verification involves manually checking data entered by Exam Aspirants, which includes large overheads with respect to time, manual labour and storage space. This leaves room for Data Redundancies and the possibility of Impersonation. Taking all these vulnerabilities into account, the web application aims to speeden up the process of Hall Ticket Verification and also curb Impersonation.

A successful implementation of this application will reduce a lot of paperwork, manual labour and time. It will also ensure that examinations are free of malpractices to a large extent.

User Personas, or the levels at which the application will be operated is as follows,

- a) Exam Administrator: Top Priority
- b) Exam Administrator: Lesser Priority

User Stories:

a) Website Users (Lesser Priority):

As an Invigilator, a lot of students are expected to get their Hall Tickets verified. On entering the Roll Number of the candidate, all details regarding that students must be auto populated in the given fields. During the first level of exams, after validation of the student, their fingerprint must be read and stored in the database. During second level of exams, the student must verify their fingerprint and an appropriate message must be displayed regarding verification.

b) Admin Users (Top Priority):

As Admin users, we must be able to choose the level of examination that is being conducted. Based on level selected web pages must display on the relevant buttons to the invigilators when they log in.

Sitemap:

Page	Items and Features
Login	<ol style="list-style-type: none">1. Username field2. Password field3. Captcha
AdminPage	<ol style="list-style-type: none">1. Option to select the level of examination2. Option to enter new examination3. Option to select duration of examination4. Option to select valid ID cards
Level One Page	<ol style="list-style-type: none">1. Roll number field2. Update Biometric button3. Buttons to read the fingerprints4. Button to take photographs.
Level Two Page	<ol style="list-style-type: none">1. Roll Number field2. Verify Candidate button

Non Functional Requirements:

- a) The application must work in all modern browsers across platforms.
- b) The application must be responsive (work well and look good on all screen sizes).

c) The application must be able to support 1000 simultaneous users.

Risks:

a) The fingerprint must be interfaced properly for the Application to run smoothly.

Software Requirements Specification

Purpose

a) Introduction

This Software Requirements Specification provides a complete description of all the functions and specifications of the software application for Examination Hall Ticket Verification and Authentication as commissioned by CDIT.

The expected audience of this document is the Informatics Division of CDIT, and the developers.

b) Scope

The Application is designed to run on the organisation server and to allow examination administrators to verify details, add biometric data, update an existing database entry, or authenticate identity using biometric data. The data will be held in a MySQL database on the organisation server.

c) Document overview

The remainder of this document is two chapters, the first providing a full description of the project for the audience. It lists all the functions performed by the Application. The final chapter

concerns details of each of the application's functions and actions in full for the software developers' assistance. These two sections are cross-referenced by topic; to increase understanding by both groups involved.

Overall description

The application encompasses numerous files and information from the Database, as well as files on the organisation server system. This system will be completely web-based, linking to remote web server from a standard web browser. An Internet connection is necessary to access the system.

a) System environment

The web site will be operated from the organisation server. When an administrator connects to the Web Server, the Web Server will pass the Administrator to the Organisation Server. The Organisation Server will then interact with the Database through Sequelize, which allows the Application to transfer data to and from a database.

b) Functional requirements definitions

Functional Requirements are those that refer to the functionality of the system, i.e., what services it will provide to the user. Nonfunctional (supplementary) requirements pertain to other information needed to produce the correct system and are detailed separately.

c) Use cases

The system will consist of CDIT Login Page, Admin Page, Level One and Level Two pages.

The login page will be based on credentials provided from the organisation to the administrator.

It will have reCAPTCHA enabled to increase security and will follow IEEE protocols to prevent attacks like DOS and SQL Injections.

The Admin Page will be accessible to those with credentials of top priority. They will have the option of adding an examination, the duration for which the application must be active, and the current level of the examination, level one or level two.

The level one page will be visible for the first iteration of an examination. The administrator will enter the roll number provided to the student and the rest of the details will be auto populated.

Then a current photograph will be taken along with 5 fingerprints of any one hand and updated in the database.

The level two page will be visible for further iterations. The administrator will enter the given roll number and upon verification with any fingerprint, the student will be authenticated.

Use Case: Access Login Page

Brief Description:

The Web Server is waiting on an administrator to login using active credentials.

Initial step-by-step description:

For this use case to be initiated, the administrator must be connected to the Internet and connected to the Web Server.

1. The Administrator connects to the Web Server.
2. The Administrator enters the valid credentials and logs in.
3. The Web Server passes the Administrator to the relevant page based on priority.

Use Case: Admin Page.

Brief Description:

The Administrator enters the examination name and duration the application will be active, and chooses level of the examination and acceptable Ids.

Initial step-by-step description:

For this use case to be initiated the Administrator must be connected to the Internet and login using credentials of top priority.

1. The Administrator enters the examination name.
2. The Administrator chooses the duration the application will be active.
3. The Administrator chooses level of examination.
4. The Administrator chooses acceptable Ids.
5. The Administrator clicks Submit.
6. The Organisation Server retains information in the database designated.
7. The Administrator logs out.

Use Case: Level One page

Brief Description:

The Administrator chooses to create a new entry for the first iteration of the examination.

Initial step-by-step description.

For this use case to be initiated the Administrator must be connected to the Internet and login using credentials of lesser priority.

1. The Administrator enters the student roll number.
2. The Administrator clicks Submit.
3. The page autofills the rest of the fields from the Database and Server Filesystem.
4. The Administrator verifies the identity.
5. The Administrator takes a new photograph.
6. The Administrator takes five fingerprints and clicks Submit.
7. If all required fields contain data the Organisation Server adds the data to the Database.
8. If a required field is empty the Organisation Server returns the form to the Administrator with a message.

9. The page refreshes to serve the next student.
10. The Administrator logs out.

Use Case: Level two page.

Brief Description:

The Administrator authenticates a fingerprint with roll number and data from the Database.

Initial step-by-step description:

For this use case to be initiated the Administrator must be connected to the Internet and login using credentials of lesser priority.

1. The Administrator enters the roll number and clicks Verify fingerprint.
2. The Student places a finger on the scanner.
3. The page returns the appropriate message if there is a match with the database details.
4. The page refreshes to serve the next student.
5. The Administrator logs out.

Non-functional requirements

There are requirements that are not functional in nature. Specifically, these are the constraints the system must work within.

- a) The web site must be compatible with all web browsers across platforms.

- b) This application will use the HTTPS protocol for Internet Security

External interface specifications

The Administrator's system must have the driver for the Mantra MFS100 Fingerprint Scanner set up.

Functional Requirements

Access Login Page

Use Case Name:	Access Login Page
Priority	Essential
Trigger	Enter URL
Precondition	Administrator is connected to the Internet
Basic Path	<ol style="list-style-type: none">1. Web Server sends the Administrator to the Organisation Server.2. The Organisation Server presents the Administrator with the Login Page.
Alternate Path	N/A
Postcondition	The Administrator is on the Login Page
Exception Path	If there is a connection failure the Organisation Server returns to the wait state
Other	-

Admin Page

Use Case Name:	Admin Page
Priority	Essential
Trigger	Top Priority credentials
Precondition	The Administrator is on the login Page and uses top priority credentials
Basic Path	<ol style="list-style-type: none">1. The Organisation Server presents the Administrator with a form.2. The Administrator fills in the form and click submit3. The Organisational Server checks to see if all required fields are not empty.4. If the required fields are not empty, the Organisation Server creates a new record in the Database.5. If any of the required fields are empty, the Organisation Server returns a message and returns the Administrator to the form.6. The Organisation Server returns the form.
Alternate Path	N/A
Postcondition	The examination record is created in the Database.
Exception Path	<ol style="list-style-type: none">1. If the connection is terminated before the form is submitted, the fields are all cleared and the Organisation Server is returned to the wait state.
Other	-

Level One Page

Use Case Name:	Level One page
Priority	Essential
Trigger	Login with less priority credentials
Precondition	The Administrator must be connected to the Internet and on the login page.
Basic Path	<ol style="list-style-type: none">1. The Organisation Server returns a form.2. The Administrator fills in the roll number field and clicks submit.3. The Organisation Server checks to see if that roll number has associated data in the database.4. If present, the Organisation Server will auto populate the rest of the fields from the database.5. The page interfaces with the webcam and fingerprint scanner to take in data.6. The administrator clicks Submit and the data is uploaded to the server file system and database.
Alternate Path	N/A
Postcondition	A record is created in the Database.
Exception Path	<ol style="list-style-type: none">1. If the connection is terminated before the form is submitted, the fields are cleared and the Organisation Server is returned to the wait state.2. If the connection is terminated after the form is submitted, but before the Administrator is returned to the refreshed

	Page, the record is created in the Database.
Other	-

Level two page

Use Case Name:	Level two page
Priority	Essential
Trigger	Login with lesser priority credentials
Precondition	The Administrator must be on the Login Page.
Basic Path	<ol style="list-style-type: none"> 1. The Organisation Server returns a form. 2. The Administrator fills in the roll number field and clicks submit. 3. The Organisation Server checks to see if that roll number has associated data in the database. 4. If present, the Organisation Server will auto populate the rest of the fields from the database. 5. The Administrator clicks verify fingerprint 6. The scanner sends the scanned fingerprint data. 7. If the data matches the database details, the form returns a confirmation, or an error message otherwise.
Alternate Path	N/A

Postcondition	The page is refreshed.
Exception Path	N/A
Other	-

Detailed non-functional requirements

Hardware: Organisation Server, Client machine, Mantra MFS100 Fingerprint Scanner

Operation System: Windows 7/Vista/8/10, Android 4.2.2 to 8.1, Ubuntu 10 and above

Internet Connection: Existing telephone lines

Code Standard: The web pages will be coded in JavaScript, CSS Bootstrap and HTML.

The backend will be coded in Node.JS

The connection to the Database will be done with Sequelize ORM.

Performance: The system should generate the records in the appropriate table of the Database
100% of the time.

System Evolution

In the future this system will be updated to allow barcode scanners instead of manually entering roll numbers. A report generated by the system of the success of verification process could be another addition to the application in the future.

