Software Requirements Specification

for

REAL TIME SUDOKO SOLVER WEB APP

Prepared by

GROUP NO.1

Naman Bansal	101803202	nbansal_be18@thapar.edu
Aadarsh Gupta	101803209	agupta3_be18@thapar.edu
Harchetan Singh Chahal	101803246	hchahal1_be18@thapar.edu

Submitted To
Mrs. Sawinder

2nd October 2020

Table of Contents

Revision History

Name	Date	Reason for Changes	Version

1. Introduction

1.1 Purpose

The purpose of this report is that it provides technical documentation regarding a web application named “Real Time Sudoku Solver Web Application”. This technical report identifies all features and lists out the functionalities and working environment in which the application can be executed. This report lists out the overall description of the application with essential features and drawbacks in the application. It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, interface and interactions with environment.

1.2 Document Conventions

Words in bold are the technical terms used in this report.

1.3 Intended Audience and Reading Suggestions

Audience who are familiar with HTML, CSS machine model implementation through Keras and TensorFlow such as Developers, testers can understand this report in much better way as this being a technical report. For Public Audiences or General Public in mind there is a Glossary at the end of the report which defines the technical terms used in the report for a better understanding of the report. The rest of the report is formatted in such a way, which starts with the documentation of SRS (software requirement specifications) and then with the application interfaces & features of the application.

Suggested reading of this technical report is to start with the overall description section, of this document which gives an overview of functionality of the product. It provides the informal requirements and is to establish a context for technical requirements specification in the next chapter. This section is for the users, project managers, marketing staff and document writers.

The third chapter, requirements specifications section, of this document is written primarily for the developers and describes in the technical terms the details of the functionality of the product.

The fourth chapter, other non- functional requirements section, of this document is written for the users and the developers as well and describes the safety and security policies.

1.4 Project Scope

This software system will be web application that help the user to extract handwritten sudoku and make user solve it on mobile application. The purpose is to enhance the experience of the user by giving hints to the user if user stuck at any stage, provides a feedback to tell user if user is solving sudoku with right approach. It also provides immediate solution to sudoku problem if user wants to solve sudoku with traditional way (using pen /pencil) to help user match his/her own answer.

Benefits:

- Helps in understanding the application (game).
- Noticing key intricacies in the game.
- Back tracking algorithm takes less memory and time compared to Dancing links algorithm.
- A solution is guaranteed (as long as the puzzle is valid).
- Solving time is mostly unrelated to degree of difficulty.

Goals:

- Saves time and paper. If player solves wrong on paper player will cut the whole problem, draw the question again and then start solving it, but in sudoku app u can reset at any point of time which will save time and resources.
- Helps as a guide or a cheat code by knowing the operating commands and strategy planning in completing the cells without any invalid moves.
- The Real time Simple Sudoku application is a freeware application, so this technical report doesn’t relate to any corporate goals or business strategies.

1.5 References

https://www.ivlabs.in/real-time-sudoku-solver.html
https://www.codeproject.com/Articles/238114/Realtime-Webcam-Sudoku-Solver
https://github.com/samuelpratt/Sudoku
http://emaraic.com/blog/realtime-sudoku-solver
https://github.com/taylorjg/sudoku-buster
https://www.youtube.com/watch?v=QR66rMS_ZfA&t=130s

We would like to express our special thanks of gratitude to our mentor Mrs. Sawinder who guided us in this wonderful project that is **Real time Sudoku solver Web app** and helped us in doing a lot of Research and we came to know about so many new things, we are really thankful to her.

2. Overall Description

2.1 Product Perspective

Simple Sudoku is Deep learning Web application implemented using Keras and TensorFlow. The main task of the algorithm is divided into two sections: Firstly, it should accurately locate the grid position in the image while taking care of problems of background clutter, scaling, translation, rotation and perspective skew. The second part of the algorithm must then locate those positions in the grid where there are numbers and recognize them with precision. Then application use backtracking algorithm to find solution of the sudoku.

Once we have acquired an image, containing the Sudoku grid we first convert it into a grayscale image, which is then the input to other preprocessing techniques so that the grid can be accurately located. The steps that we took are discussed below:

1) Smoothing the image and Adaptive thresholding.
2) Isolating the grid from background clutter of the image.

In backtracking, we first start with a sub-solution and if this sub-solution doesn't give us a correct final answer, then we just come back and change our sub-solution. We are going to solve our Sudoku in a similar way. The steps which we will follow are:

- If there are no unallocated cells, then the Sudoku is already solved. We will just return true.
- Or else, we will fill an unallocated cell with a digit between 1 to 9 so that there are no conflicts in any of the rows, columns, or the 3x3 sub-matrices.
- Now, we will try to fill the next unallocated cell and if this happens successfully, then we will return true.
- Else, we will come back and change the digit we used to fill the cell. If there is no digit which fulfils the need, then we will just return false as there is no solution of this Sudoku.

2.2 Product Features

1) **Register of the player**
Those who are new to app are required to register and then login to get benefit of it.

2) **Login in the web app:**
For those who have been already registered, can enter their valid e-mail and can get benefit of it.

3) S**can the Sudoku:**
It is used to scan the sudoku from newspaper or from other sources. It uses OpenCV library and deep learning models to interpret edges, matrix along with digit recognition. At end, the scanned sudoku is given to web app for further computation and utility for fulfillment of purpose of the app.

4) **Write in the cell:**
Select or active the cell in which you want to write. Write digit in the active or selected cell, the digit you think would lead to the solution.

5) **Validate**
It is used to check whether the value entered by the user and the value that should have been present are same or not. It returned true if the values are same otherwise false

6) **See the hint:**
Hint is used by user to get the value that should be placed in cell of the sudoku solver matrix. The cell is initially a blank cell which is filled afterwards by using hint.

7) **Show the overall solution of sudoku:**
Completely not getting the clue and of sudoku as it seems to be difficult, you can get complete solution of sudoku and can interpret the tricks the problem was having to develop your mind accordingly.

8) **Reset the sudoku:**
Reset your problem and start again in case u think your approach in finding solution seems wrong.

2.3 User Classes and Characteristics

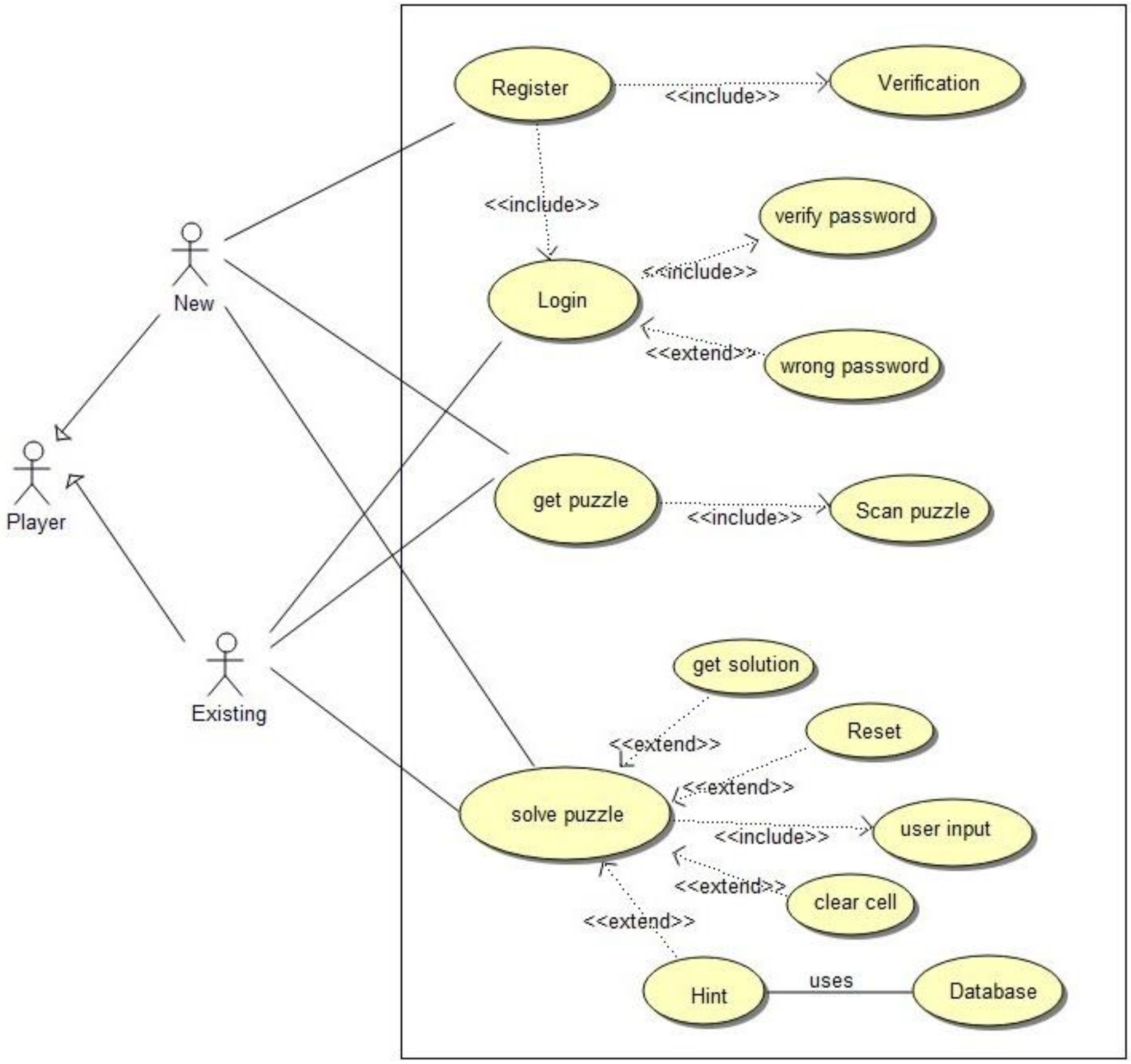

Figure 1: UseCase

1) **Register:** The user will first register to the website to be a authorized user. He/she will provide the personal information like username, email and password. After creating an account, there will an email verification by which the admin of the webpage will get to know whether the information provided by the user are right or not

2) **Login:** The user have to register first before getting login. He/she can't login without registering. Here the credentials of user are verified with the one when he provided during the registration. If credentials match, user is given rights to access the web page whereas if user fails, then the wrong information message will be prompted.

3) **get puzzle:** To get the puzzle, user has to point webcam at a sudoku puzzle. OpenCV library and deep learning models are used to interpret edges, matrix along with digit recognition. Once the web app recognizes the puzzle, stop the webcam. At end, the scanned sudoku is given to web app for further computation and utility for fulfillment of purpose of the webapp.

4) **solve puzzle:** To solve the puzzle completely, one can use the following functionalities

   - USER INPUT: Here user provides the input to a cell, and it is validated with the actual value that should be present using sudoku's solution present in database. If both value matches, the digit's font color changes to green specifying that entered digit is correct and to red if it's wrong.

   - HINT: If user gets struck in the puzzle and unable to get which cell to enter next and with which value, he/she can use hint functionality to get the value of certain cell in the sudoku.

   - SOLUTION: If user wants to see whole solution at once, he/she can use solution functionality.

   - CLEAR CELL: It is used to clear the value in certain cell. If user feels that he has filled the incorrect value in the cell then he can clear that cell using this functionality

   - RESET: It clears all the values entered by the user.

2.4 Operating Environment

The hardware requirements will be that the user needs 100 MB or more of storage so that the Web app can run smoothly, the phone should have more than 2 GB of RAM.

Operating system	**Web Browser**
Windows 7, 8.1, Server 2008, Server 2012, Server 2016	Microsoft Internet Explorer 11 Mozilla Firefox 7.0 or later (JavaScript and Cookies enabled)
Windows 10	Microsoft Internet Explorer 11 Microsoft Edge Mozilla Firefox 7.0 or later (JavaScript and Cookies enabled)

Mac OS	Mozilla Firefox 7.0 or later (recommended) (JavaScript and Cookies enabled)
Linux	Mozilla Firefox 7.0 or later (recommended) (JavaScript and Cookies enabled)

2.5 Design and Implementation Constraints

The design constraints are mainly depending upon the operating system or the web browser on which it will get uploaded. Since most of the user in India are Firefox browser uses so our web app will be catering to their needs so our design will according latest browser version. The Sudoku Solver app would be written HTML, CSS framework and backend will be done using Django or Flask. The system must provide a capacity for parallel operation and system design should not introduce scalability issues.

2.6 User Documentation

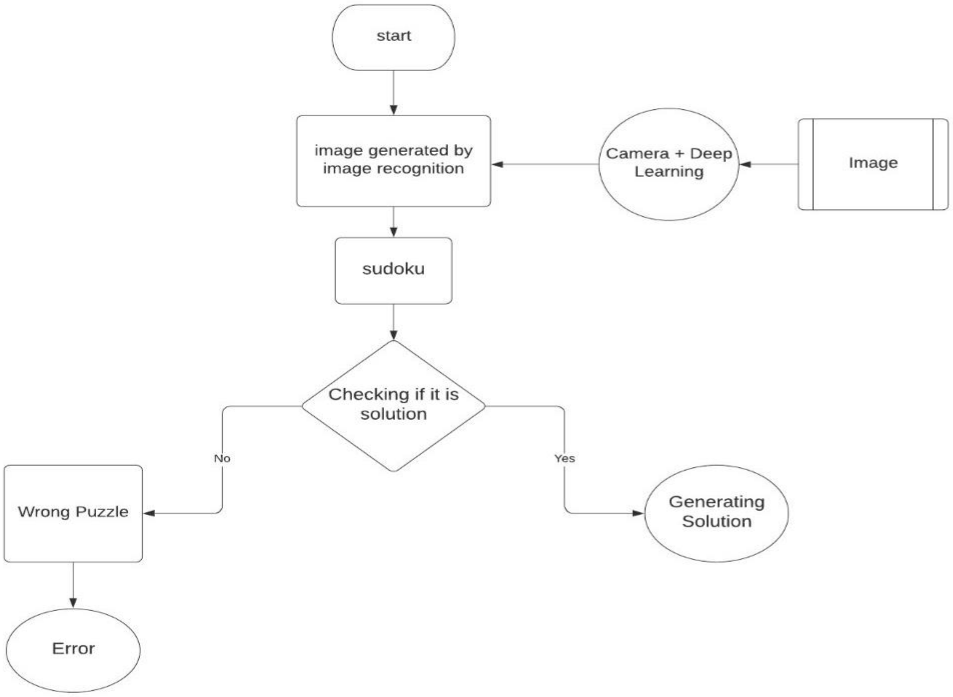

Figure 2: Flowchart 1

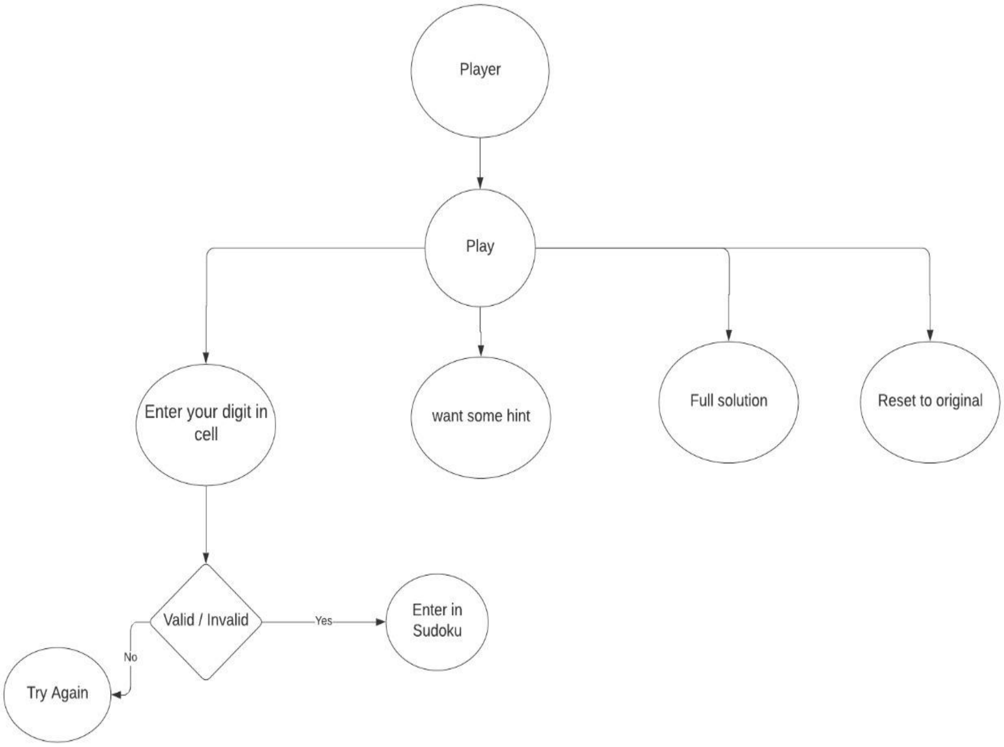

Figure 3: Flowchart 2

1) The above flowchart explains user how the app works, it is written in simple language to help users to understand it easily.

2) In short user need to login into the app after registration.Then, user will take picture from camera of sudoku basically scan sudoku and after tht sudoku will get on Web app‘s interface. Now, user can solve sudoku and enjoy the hint,reset, etc features of the app.

3) The elaborate user guidance document will be provided after completion of the project .

2.7 Assumptions and Dependencies

Assumption 1: The users are willing to use app and feel comfortable and ease in using mobile phones.

Assumption 2: There should not be immediate answers provided along with handwritten sudoku. Like in newspaper or daily magazines the Sudoku problems are given at every weekends and solution are given in next consecutive weekend.

Assumption 3: The sudoku problem should have answer. The filled cells of the Sudoku should have been correctly filled in accordance to the rules of the game, or the backtracking algorithm would not be able to find the solution of problem.

3. System Features

The sudoku solver web app helps you to scan problem to mobile interface and provide additional functionalities to enhance experience of problem solving. The functionalities include hint system (if the user gets stuck at any stage of problem), providing immediate full solution, reset, etc. The web app uses deep learning model which is trained and implemented using Kera's and TensorFlow

3.1 Sudoku recognition

3.1.1 Description and Priority

Main feature without which other features are not worthy. The feature is to scan physical or handwritten sudoku to your app and enjoy the features provided by the app.

3.1.2 Stimulus/Response Sequences

For user need nothing, its developer who will train dataset only user need to click sudoku with camera and should click with appropriate angle.

3.1.3 Functional Requirements

Test data for image recognition especially for digit recognition. We also need good internet for computational capacity for implementation of algorithm

3.2 Reset Sudoku

3.2.1 Description and Priority

Every weekend newspaper has sudoku problems printed in it. Curious user is tended to do mistakes while solving the problem. If a major mistake is made while solving the problem, they had to draw the sudoku again which wastes both resources and time. By scanning the sudoku to mobile interface, user could not only save time but could have iterative interface to solve.

3.2.2 Stimulus/Response Sequences

The entries entered from user will get 0. User is not needed to provide anything for this feature but should use carefully. If clicked accidently, all of the user input will get to 0, and user had to think logic again. In this way if user did not used it wisely it can also become demerit of the system.

3.2.3 Functional Requirements

Database to store the sudoku that has been computed using Image recognition Model. As it uses machine or deep learning algorithm, we need the system of good computational capacity to implement these algorithm.

3.3 Helping with Hint

3.3.1 Description and Priority

Hint is used to solve the naked single, which shows the blank cell which is color marked. The feature helps to guide user to solution. If the user is stuck at any stage it can use this feature and can move further. Basically, it's good to give hints rather full solution because hints help user to solve puzzle by themselves and develop their minds and enhance their IQ level.

3.3.2 Stimulus/Response Sequences

In Hint system user need not to do anything. Hint system works by seeing corresponding selected cell in solution database and give the digit to user. The solution of sudoku is driven from Backtracking algorithm. Only one thing user should take care of is that the sudoku scanned should be valid one.

3.3.3 Functional Requirements

Database to store the solved sudoku that has been computed using backtracking Algorithm.

3.4 Immediate Full Solution

3.4.1 Description and Priority

The feature helps user to get immediate solution of the problem. This thing help user to not wait for next weekend newspaper to match their answer with solution if user tends to solve on paper or on system.

3.4.2 Stimulus/Response Sequences

In Immediate solution, the solution calculated from backtracking is provided. User need nothing to do for this feature. Only one thing user should take care of is that the sudoku scanned should be valid one.

3.1.3 Functional Requirements

Database to store the solved sudoku that has been computed using backtracking

Algorithm

4. External Interface Requirements

4.1 User Interfaces

The software provides good graphical interface for the user and the user can operate on the system, performing the required task like scanning, solving, getting help or getting the full solution.

4.2 Hardware Interfaces

Processor: Pentium(R) Dual-core CPU
Hard Disk: 40GB
RAM: 256 MB or more

- The main requirement for sudoku solver web app is camera to scan the sudoku and get it as input to the Web app.
- Then requirements for playing Simple Sudoku are mouse or Track pad.
- Other requirements such as processor, memory which is already mentioned under Operating environment section.

4.3 Software Interfaces

The software package is developed using HTML and CSS, backend is done by Django and the Deep Learning model is trained using Kera's and implemented using TensorFlow.

Operating System: Window XP, Windows 7 and higher version
Language: HTML, Flask, CSS, HTML, javaScript
Database: SQLAlchemy, Mysql(Local Machine) , postgresql(deploying)

4.4 Communications Interfaces

Simple Sudoku does not have any communication interfaces, communication interface resembles such as google talk or yahoo messenger. It does not require any messaging protocol such as HTTP, FTP protocols. It only uses email id for registration purpose only.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The performance of the application is measured while it is played which means it evaluated on real time performance. Average page load (from a user perspective) must be less than 500 milliseconds. Slowest page load cannot take more than 4 seconds. The app must be available more than 99.5% of the time, on average

5.2 Safety Requirements

Code safety ensures whether software is reliable and safe to use. Being an application, which contributes for the welfare of the society and also not demanding any other personal information except email-id for login, therefore any specific safety measure was not encountered.

5.3 Security Requirements

Security systems need database storage just like many other applications. However, the special requirements of the security market mean that vendors must choose their database partner carefully.

5.4 Software Quality Attributes

1) **Availability:** Order can be booked within whole Patiala city and system will be available 24x7.

2) **User friendly**: System should be easily used by customer.

3) **Reliability**: The ability of the system to behave consistently in a user acceptable manner when operating within the environment for which the system was intended. As mentioned earlier the program is estimated to be 98% successful based on case studies. The actual reliability can be confirmed once the application is on the field.

4) **Efficient:** System should be efficient that it won’t get hang if heavy traffic of order in placed.

5) **Maintainability**: The application will be easy to extend. The code is written in a way that it favors implementation of new functions

Appendix A: Glossary

HTML	HTML is the standard markup language for Web pages. With HTML you can create your own Website.
CSS	CSS is the language we use to style an HTML document. CSS describes how HTML elements should be displayed.
Django	Django is a web development framework that assists in building and maintaining quality web applications.
Flask	It is lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications.
Includes	Has the appropriate constraint in it
Connects	Links this requirement with another
TensorFlow	TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks
Keras	Kera's is a deep learning API written in Python, running on top of the machine learning platform TensorFlow.

Appendix B: Analysis Models

An Analysis model such as an Use case UML diagram has been included in the Software Functions under section 2.3 for analysis of the application Simple Sudoku.

CLASS DIAGRAM

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of objectoriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

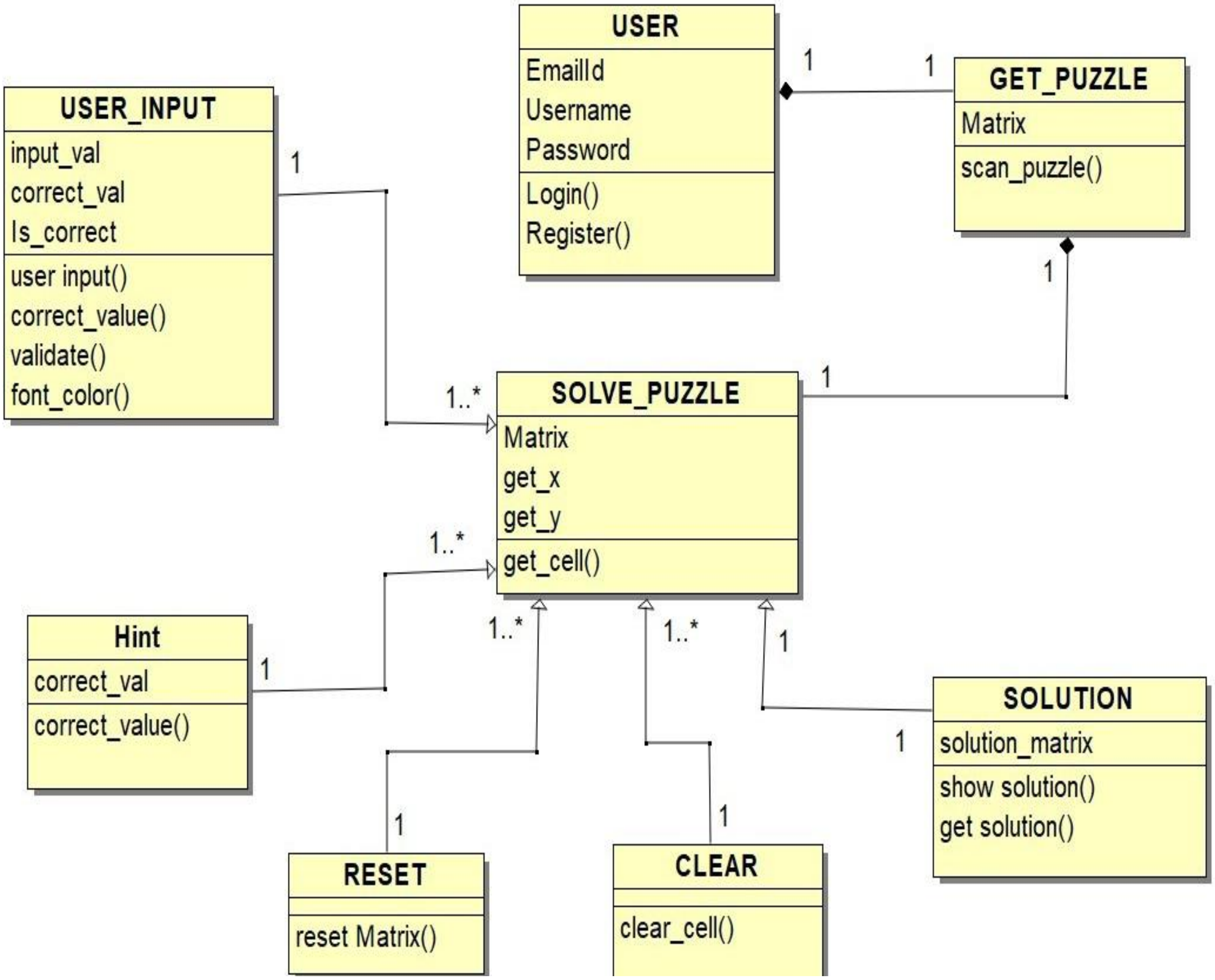

Figure 4:Class diagram

ACTIVITY DIAGRAM

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system.

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc

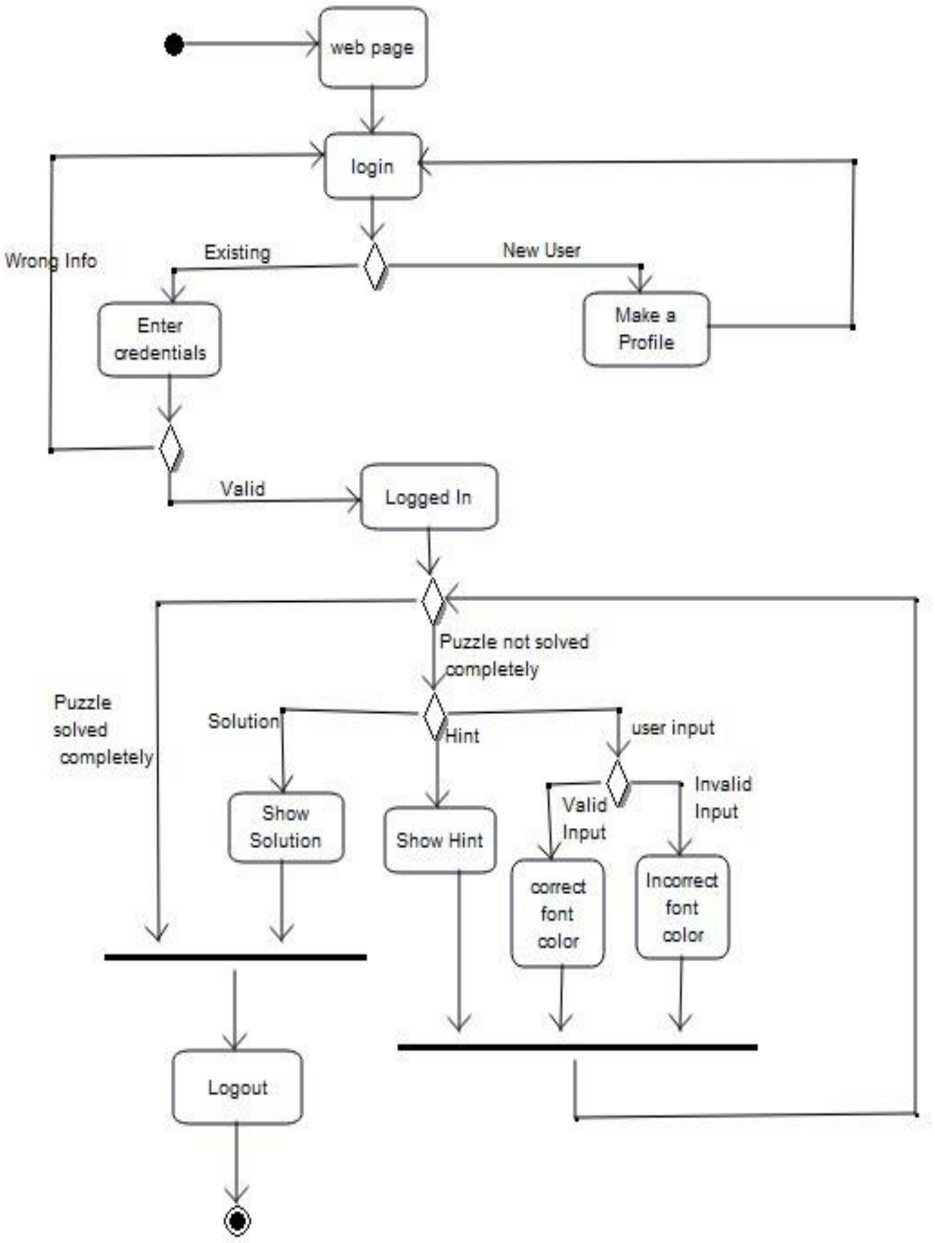

Figure 5: Activity Diagram

DATA FLOW DIAGRAM

Also known as DFD, Data flow diagrams are used to graphically represent the flow of data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation.

LEVEL 0 DATA FLOW DIAGRAM

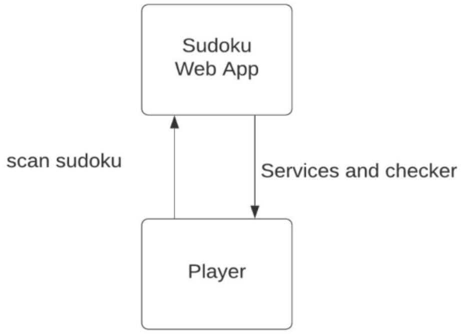

Figure 6: DFD level 0

LEVEL 1 DATA FLOW DIAGRAM

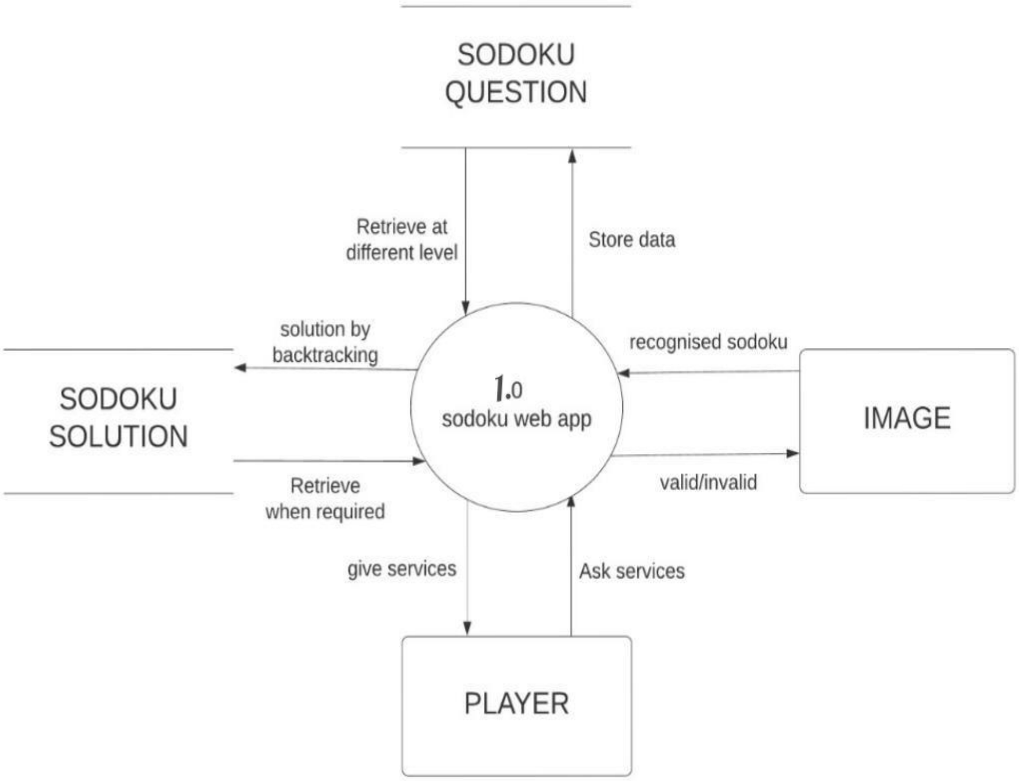

Figure 7: DFD level 1

LEVEL 2 DATA FLOW DIAGRAM

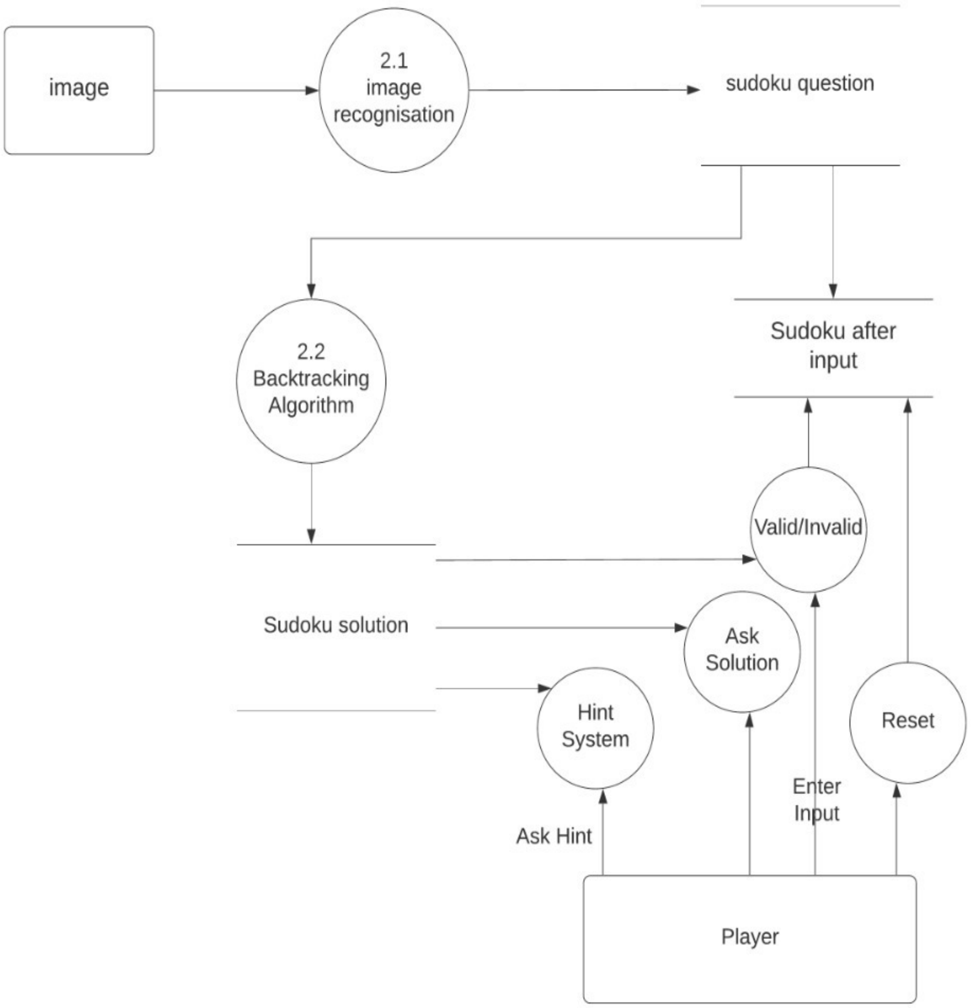

Figure 8: DFD level2

STATE DIAGRAM

State Machine Diagrams (or sometimes referred to as state diagram, state machine or state chart) show the different states of an entity. State machine diagrams can also show how an entity responds to various events by changing from one state to another. State machine diagram is a UML diagram used to model the dynamic nature of a system.

Initial state	The initial state symbol is used to indicate the beginning of a state machine diagram.
Final state	This symbol is used to indicate the end of a state machine diagram.
Decision box	It contains a condition. Depending upon the result of an evaluated guard condition, a new path is taken for program execution.
Transition	A transition is a change in one state into another state which is occurred because of some event. A transition causes a change in the state of an object.
State box	It is a specific moment in the lifespan of an object. It is defined using some condition or a statement within the classifier body. It is used to represent any static as well as dynamic situations.

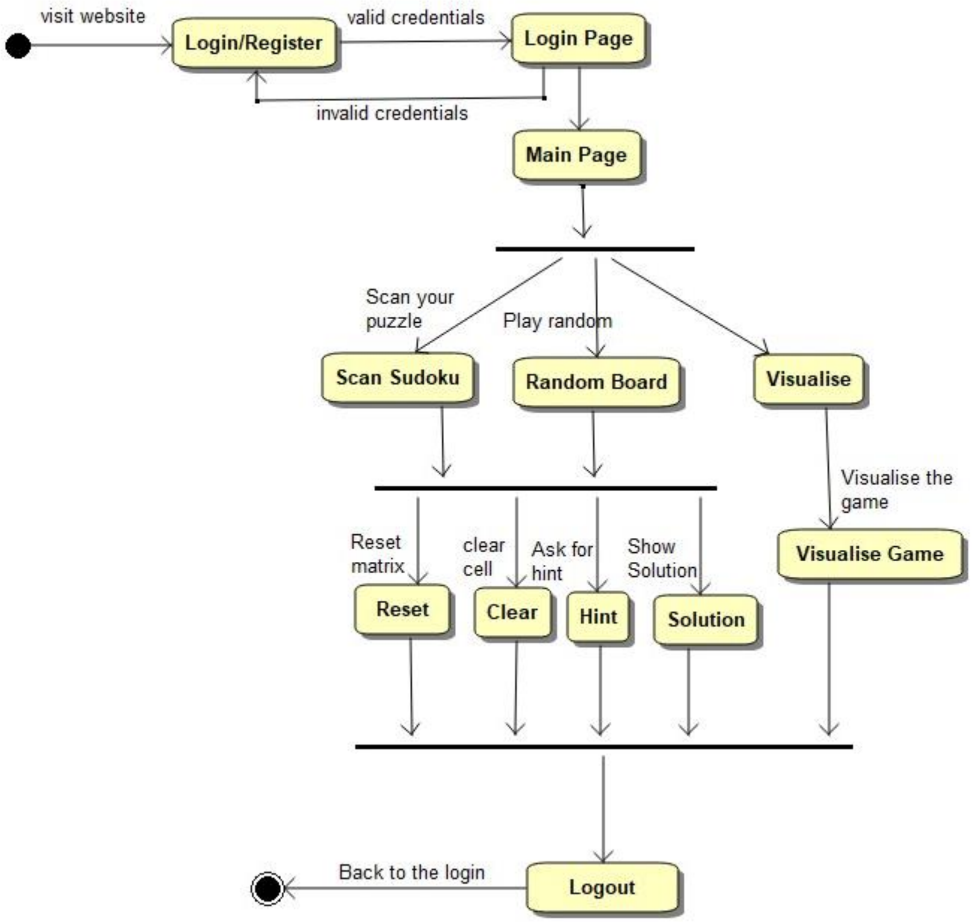

Figure 9: State Diagram

SEQUENCE DIAGRAM

Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.

Actors	An actor in a UML diagram represents a type of role where it interacts with the system and its objects. It is important to note here that an actor is always outside the scope of the system we aim to model using the UML diagram.
Lifelines	A lifeline is a named element which depicts an individual participant in a sequence diagram. So basically each instance in a sequence diagram is represented by a lifeline. Lifeline elements are located at the top in a sequence diagram. The standard in UML for naming a lifeline follows the following format – Instance Name : Class Name
Messages	Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline. We represent messages using arrows. Lifelines and messages form the core of a sequence diagram.
Guards	To model conditions we use guards in UML. They are used when we need to restrict the flow of messages on the pretext of a condition being met. Guards play an important role in letting software developers know the constraints attached to a system or a particular process.

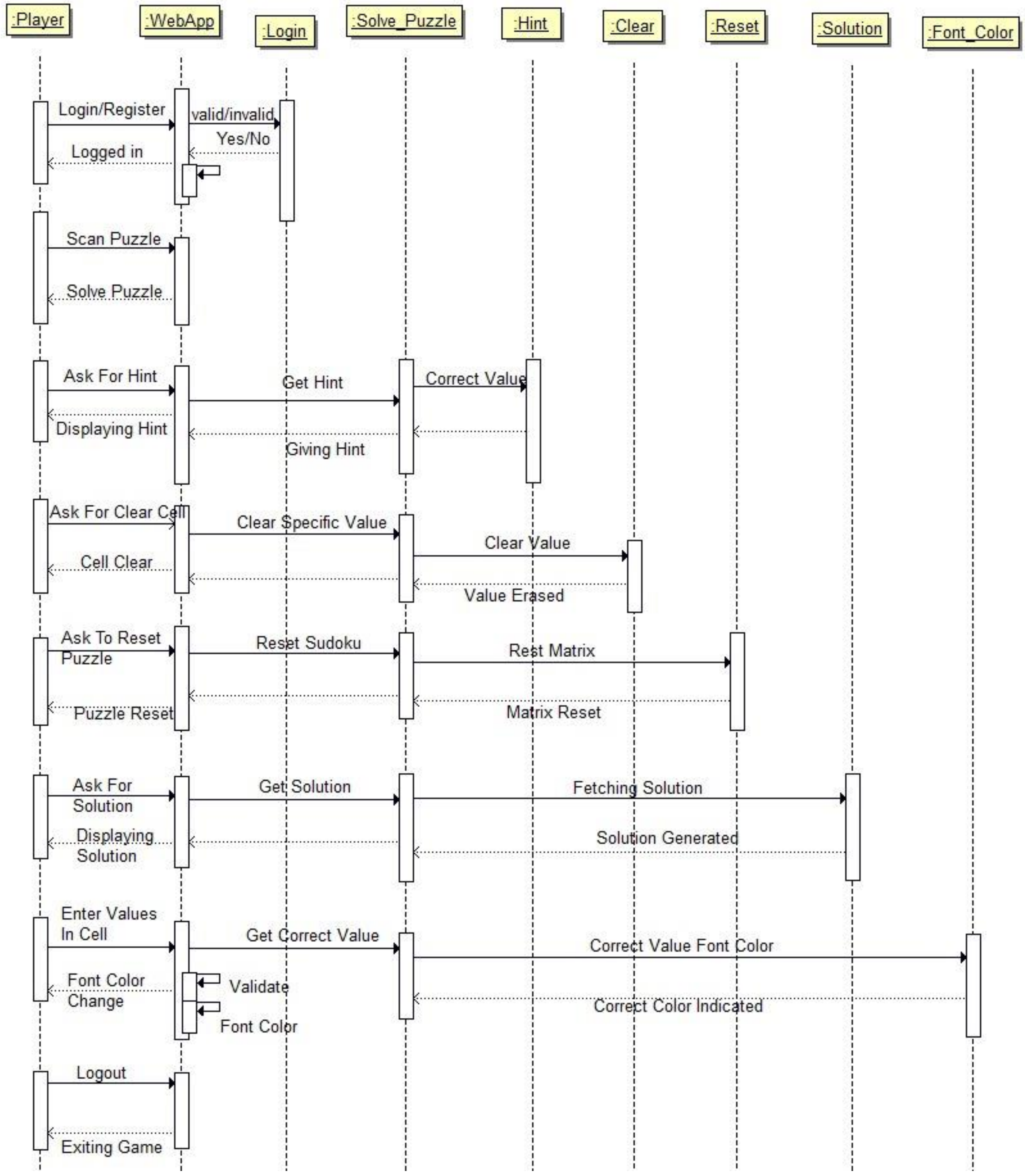

Figure 10: Sequence Diagram

COLLABORATION DIAGRAM

A collaboration diagram, also known as a communication diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML).

Actors	In the collaboration diagram, the actor plays the main role as it invokes the interaction. Each actor has its respective role and name. In this, one actor initiates the use case.
Objects	The representation of an object is done by an object symbol with its name and class underlined, separated by a colon
Links	The link is an instance of association, which associates the objects and actors. It portrays a relationship between the objects through which the messages are sent. It is represented by a solid line. The link helps an object to connect with or navigate to another object, such that the message flows are attached to links
Messages	It is a communication between objects which carries information and includes a sequence number, so that the activity may take place. It is represented by a labeled arrow, which is placed near a link. The messages are sent from the sender to the receiver, and the direction must be navigable in that particular direction. The receiver must understand the message.

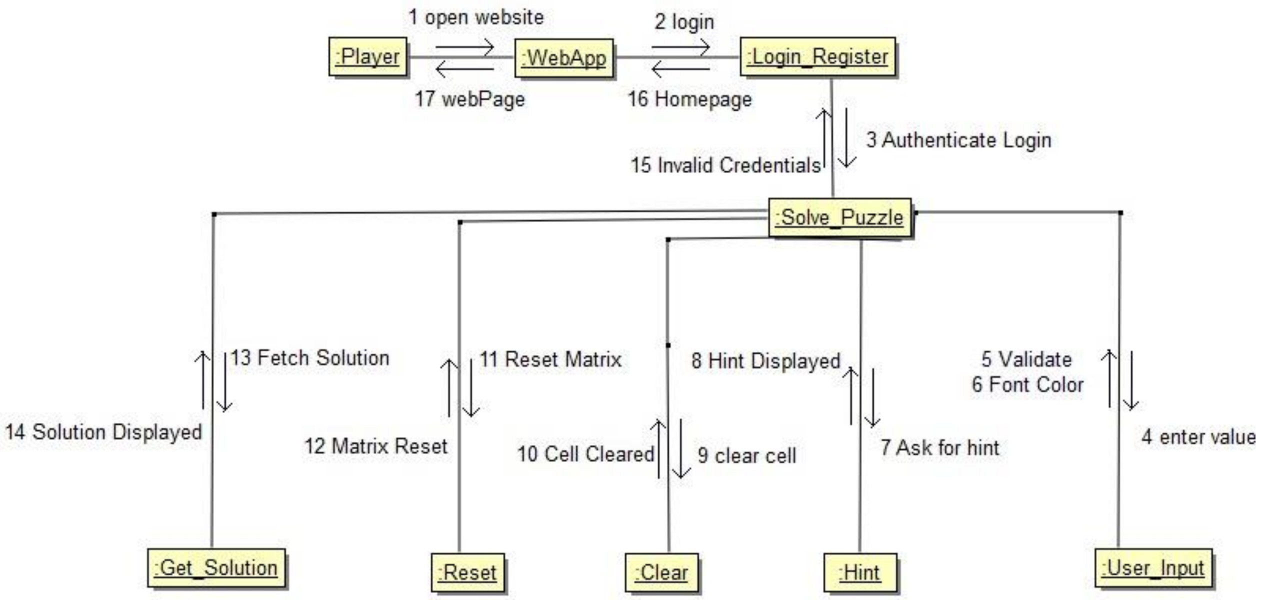

Figure 11: Collaboration Diagram

COMPONENT DIAGRAM

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development.

Component	A component is a logical unit block of the system, a slightly higher abstraction than classes. It is represented as a rectangle with a smaller rectangle in the upper right corner with tabs or the word written above the name of the component to help distinguish it from a class.
Interface	An interface (small circle or semi-circle on a stick) describes a group of operations used (required) or created (provided) by components. A full circle represents an interface created or provided by the component. A semi-circle represents a required interface, like a person's input.
Dependencies	Draw dependencies among components using dashed arrows.
Port	Ports are represented using a square along the edge of the system or a component. A port is often used to help expose required and provided interfaces of a component

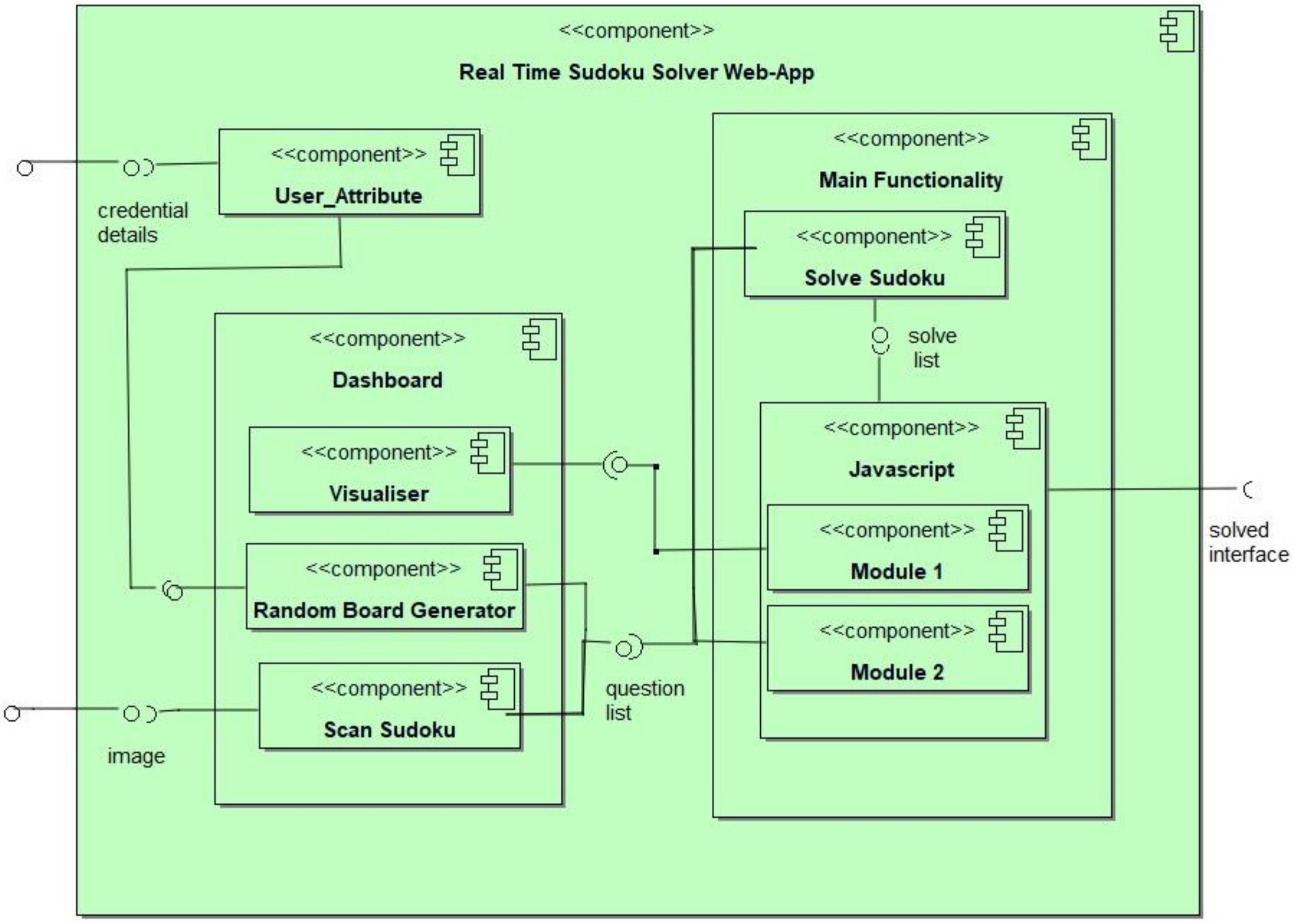

Figure 12: Component Diagram

DEPLOYMENT DIAGRAM

A deployment diagram is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them.

Nodes	A node, represented as a cube, is a physical entity that executes one or more components, subsystems or executables. A node could be a hardware or software element.
Artifacts	Artifacts are concrete elements that are caused by a development process. Examples of artifacts are libraries, archives, configuration files, executable files etc.
Communication Association	This is represented by a solid line between two nodes. It shows the path of communication between nodes.

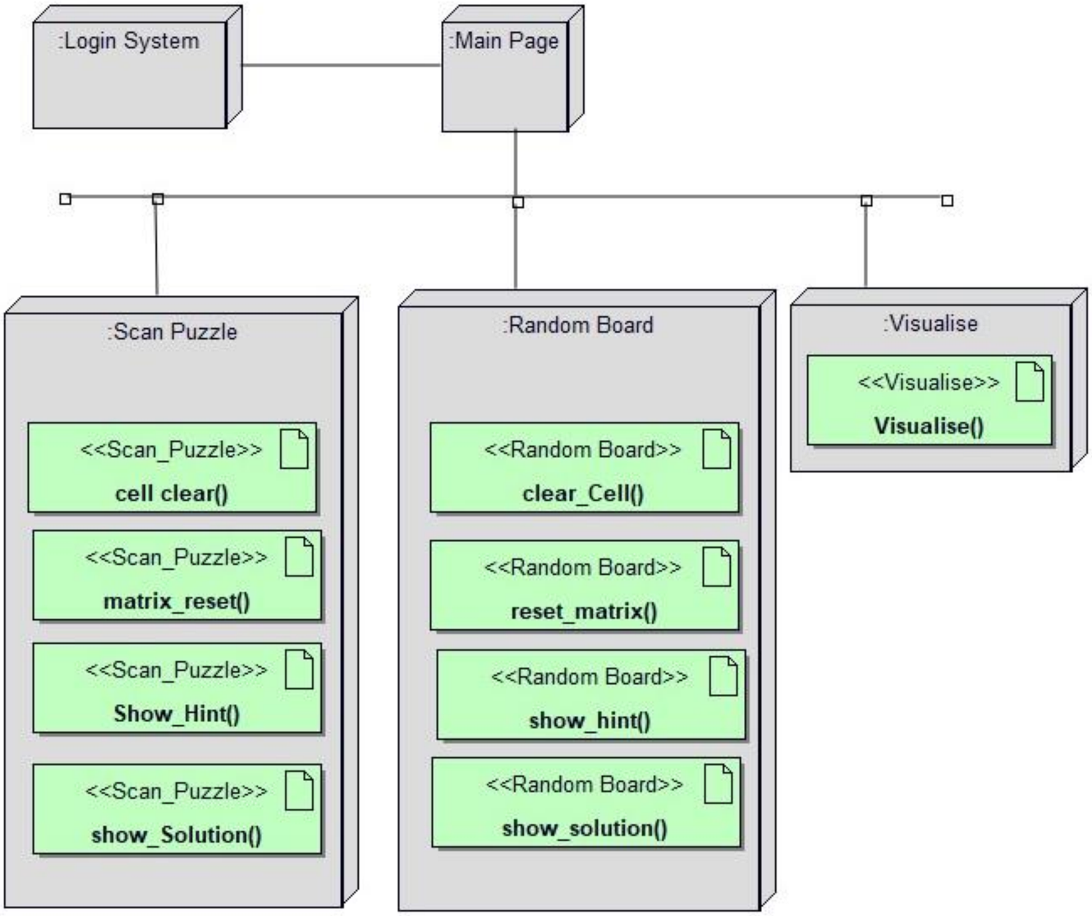

Figure 13:Deployment Diagram

Appendix C: Test Cases

Test Case#: 1 **Test Case Name:** Dashboard

System: Sudoku Solver **Subsystem:** Dashboard

Designed by: Naman Bansal **Design Date:** 11-11-2020

Executed by: Aadarsh Gupta **Execution Date:** 17-11-2020

Short Description: Used bootstrap template. Features and joining with other module is done through flask. At back-end, python script are enacted. We also used flask_mail library to activate contact us form

Pre-Conditions:
The user has not registered himself with web app

Step	Action	Expected System Response	Pass / Fail	Comment
1	Click 'Random generator' button	The python script along with Javascript gets executed and u get random question to solve.Note that every time different question must be generated	PASS	Buttons are used to connect one page to other.
2	Click 'Visualizer' button	The visualizer code window gets open	PASS	Buttons are used to connect one page to other.
3	Click 'Scan' button and the sudoku board does not get scanned properly.	When the sudoku does not get scanned properly,we should get to know.for this message gets flashed to warn user to redo the process.	PASS	Exception handling is must
4	Click on 'Scan' button and the sudoku button gets scanned properly	The python script should get run and should be moved to next page	PASS	Success message is meant to aware user

Post-Conditions:
The user has not registered himself with web app

Test Case#: 2	**Test Case Name:** Random Solver Board
System: Sudoku Solver	**Subsystem:** Random Board
Designed by: Aadarsh Gupta	**Design Date:** 16-11-2020
Executed by: Aadarsh Gupta	**Execution Date:** 27-11-2020

Short Description: It provides user interface to solve sudoku problem. User is provided with various functionalities like hint for getting some random hint, solution for getting sudoku solution all at once, clear for clearing value filled in some specific cell and reset for clearing all the values filled by user .

Pre-Conditions:
User has created a session and can access features of the app

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Click on 'Hint' button	Fills one of the unfilled cells with correct value and the cell turns orange. Note these cells cannot be edited further.	PASS	Note these cells cannot be edited by user as they already are entered with correct value.
2	Click on 'Get Solution'	Fills all cells that are empty with correct value	PASS	When user is clueless and does not find any approach.
3	Change of color	If the value entered in cell is found incorrect according to rule of Sudoku, the cell is highlighted with red color.	PASS	Color highlighting is just to fascinate user and make it more user friendly.
4	Click on 'Reset' button	The values entered by user will get reset to empty cell.	PASS	All the cells filled by user, are cleared.
5	Click on 'Clear' button	This will clear the value from selected cell.	PASS	Specific cells are cleared
6	When all cells are filled in Sudoku	If all cells are entered with correct value, then message gets flashed and the board cells (entered by user) are highlighted with green color.	PASS	The objective of project ends here.
7	When random board generator module is opened	Every time different sudoku board (question) should be generated	PASS	Different board should be generated as per script
8	Click on 'Log out' button	The session of logged in user will get finish	PASS	Session making can be complicated.
9	Click on 'Dashboard' button	This will revert back to dashboard page.	PASS	

Post-Conditions:
User will click on one of the button to enjoy one of the feature of sudoku web app

Test Case#: 3 **Test Case Name:** Scan Sudoku Board

System: Sudoku Solver **Subsystem:** Sudoku Scan board

Designed by: Harchetan Singh **Design Date:** 16-11-2020

Executed by: Aadarsh Gupta **Execution Date:** 27-11-2020

Short Description: It provides user interface to solve sudoku problem. The problem is scanned from hard copy to computer interface . User is provided with various functionalities like hint for geting some random hint, solution for getting sudoku solution all at once, clear for clearing value filled in some specific cell and reset for clearing all the values filled by user .

Pre-Conditions:
User has created a session and can access features of the app

Step	Action	Expected System Response	Pass/ Fail	Comment
1	Click on ‘Hint’ button	Fills one of the unfilled cells with correct value and the cell turns orange. Note these cells cannot be edited further.	PASS	Note these cells cannot be edited by user as they already are entered with correct value.
2	Click on ‘Get Solution’	Fills all cells that are empty with correct value	PASS	When user is clueless and does not find any approach.
3	Change of color	If the value entered in cell is found incorrect according to rule of Sudoku, the cell is highlighted with red color.	PASS	Color highlighting is just to fascinate user and make it more user friendly.
4	Click on ‘Reset’ button	The values entered by user will get reset to empty cell.	PASS	All the cells filled by user, are cleared.
5	Click on ‘Clear’ button	This will clear the value from selected cell.	PASS	Specific cells are cleared
6	When all cells are filled in Sudoku	If all cells are entered with correct value, then message gets flashed and the board cells (entered by user) are highlighted with green color.	PASS	The objective of project ends here.
7	When scan module is opened	Board generated should have same values as per scanned image	PASS	Digit Recognition will be used for predicting the values entered
8	Click on ‘Log out’ button	The session of logged in user will get finish	PASS	Session making can be complicated.
9	Click on ‘Dashboard’ button	This will revert back to dashboard page.	PASS	

Post-Conditions:
User will click on one of the button to enjoy one of the feature of sudoku web app

We deployed the model on Heroku platform. You can use https://real-time-sudoku-solver-api.herokuapp.com/ to run it on your local machine