

SRS OF SUDOKU APP

1. Introduction

- 1.1 Purpose
- 1.2 Document Conventions
- 1.3 Intended Audience and Reading Suggestions
- 1.4 Project Scope
- 1.5 References

2. Overall Description

- 2.1 Product Perspective
- 2.2 Product Features
- 2.3 User Classes and Characteristics
- 2.4 Operating Environment
- 2.5 Design and Implementation Constraints

3. System Features

- 3.1 Functional Requirements

4. External Interface Requirements

- 4.1 User Interfaces
- 4.2 Hardware Interfaces
- 4.3 Software Interfaces

5. Nonfunctional Requirements

- 5.1 Performance Requirements
- 5.2 Safety Requirements
- 5.3 Security Requirements
- 5.4 Software Quality Attributes

6. Use Cases

- 6.1 Use Cases Table

1. Introduction

1.1 Purpose

The purpose of this document is to build a sudoku mobile game for Android OS with single and multiplayer modes.

1.2 Document Conventions

This document uses the following conventions:

DB	Database
ER	Entity Relationship

1.3 Intended Audience and Reading Suggestions

This project is ready single / multiplayer sudoku game. The project is aimed at an audience who wants to have fun playing the famous logic game

1.4 Project Scope

The main purpose of the game is entertaining people . The system is based on a relational database with its users management and recording scores. We wanted to make that game not only fun, brain development, but also challenging. Above all, we hope to provide a comfortable user experience along with the best design patterns.

1.5 References

<https://play.google.com/store/apps/details?id=com.maksym.dovhan.sudoku.plus>

2. Overall description

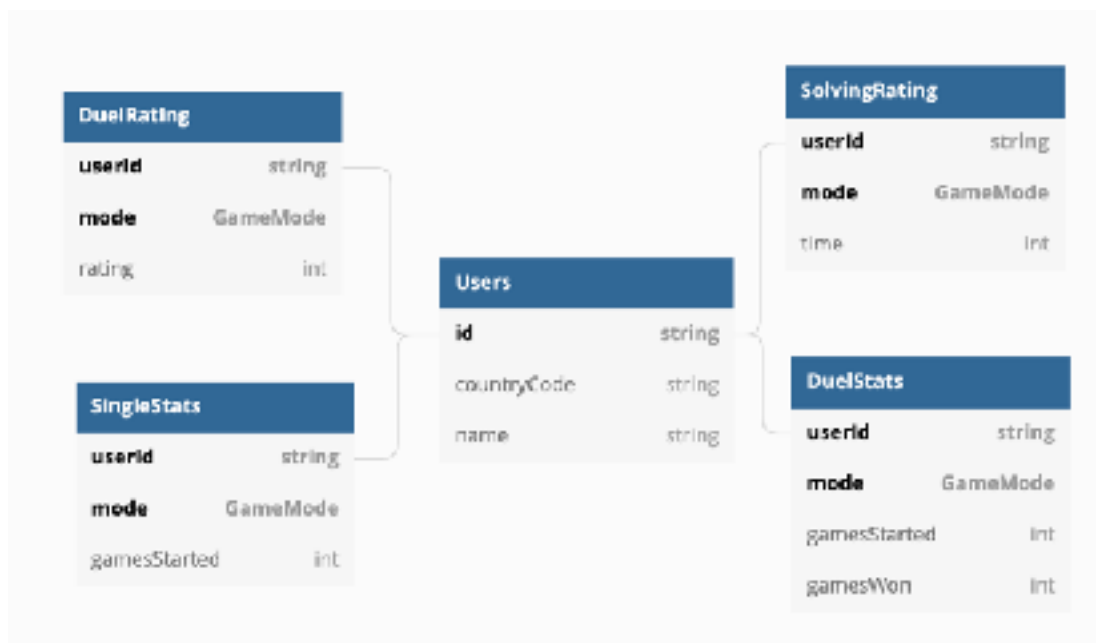
2.1 Product Perspective

The **Sudoku+** Game Database System stores the following information:

- Information about user: credentials for login and profile info.
- World rating with user id, difficulty, rating type and time.

2.2 Product Features

The major features of airline database system as shown in below entity–relationship model (ER model):



2.3 User Class and characteristic

Users will be able to play both in solo mode and with a friend at speed. Solo mode is divided into competitive and free for a more relaxed game without a timer. Also, the result, namely the speed of completion of a successful Sudoku solution, will be added to the database and displayed in the world ranking

List of actions available to the user:

- Registration
- Authorization
- Selection of the game mode
- Setting up your own profile

2.4 Operating Environment

Operating environment for the sudoku+ game system is as listed below.

- client/server system
- Operating system: Android.
- database: mssql database
- platform: C#, Dart, Flutter

2.5 Design and implementation constrains

- The global validation schema for user input fields
- SQL commands for above queries / applications and game results
- Implemented the databased using a centralised database management system.

3. System Features

Description and priority

The sudoku game is about entertainment. It allows to play logical sudoku game and spent time having fun as solo as with friends. The game has middle priority as it could be very popular, but at the same time not very useful for the global.

Response Sequences

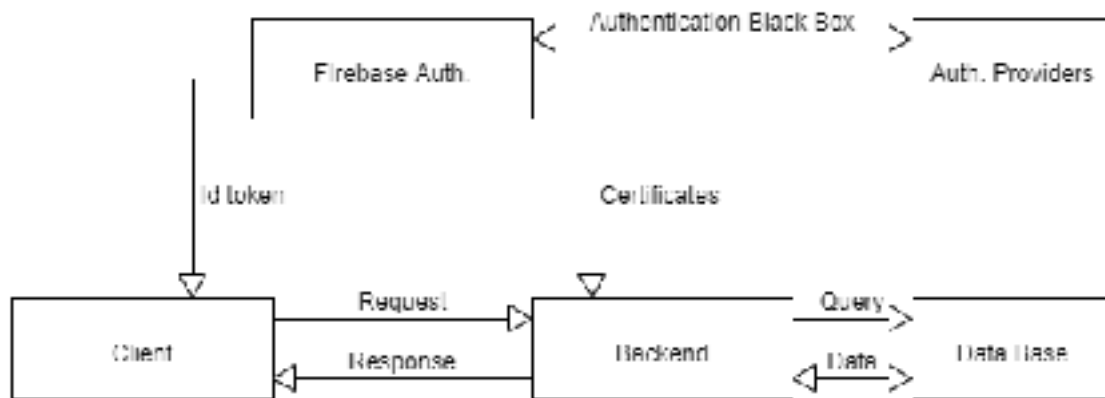
1. Select game mode
2. If multiplayer — find opponent
3. Start playing
4. Cancel / Finish
5. Send result to database
6. Show results of each player in world rating

Distributed Database

Distributed database implies that a single application should be able to operate transparently on data that is spread across a variety of different databases and connected by a communication network as we have option to filter by country.

Client / server

Represented with architecture diagram.



4. External Interface Requirements

4.1 User Interfaces

- Front-end software: Dart, Flutter
- Back-end software: C#, MSSQL

4.2 Hardware Interfaces

- Android OS

4.3 Software Interfaces

Following are the software used for the sudoku game application.

Software used	Description
Operating system	We have chosen Android operating system for its best support and user-friendliness
Database	To save the game results, user records we have chosen MSSQL Database
Dart , Flutter, C#	To implement the project we have chosen Dart, Flutter and C# technologies stack.

5. Nonfunctional Requirements

5.1 Performance Requirements

The steps involved to perform the implementation of airline database are as listed below.

A) The E-R Diagram constitutes a technique for representing the logical structure of a database in a pictorial manner. This analysis is then used to organize data as a relation, normalizing relation and finally obtaining a relation database.

- **Entities:** Which specify distinct real-world items in an application
- **Properties / Attributes:** Which specify properties of an entity and relationships.
- **Relationships:** Which connect entities and represent meaningful dependencies between them.

B) Normalization:

The basic objective of normalization is to reduce redundancy which means that information is to be stored only once. Storing information several times leads to wastage of storage space and increase in the total size of the data stored. If a database is not properly designed it can give rise to modification anomalies. Modification anomalies arise when data is added to, changed or deleted from a database table. Similarly, in traditional databases as well as improperly designed relational databases, data redundancy can be a problem. These can be eliminated by normalizing a database.

Normalization is the process of breaking down a table into smaller tables. So that each table deals with a single theme. There are three different kinds of modifications of anomalies and formulated the first, second and third normal forms (3NF) is considered sufficient for most practical purposes. It should be considered only after a thorough analysis and complete understanding of its implications.

5.2 Safety Requirements

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.

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

- **Availability:** The game should be available for users having stable network connections as sudoku game scheme generates on server side
- **Correctness:** The game statistic should be sent to the server and added to the database with correct data.
- **Maintainability:** The game world rating should be correct at the moment when user sends request.
- **Usability:** The game difficulties and worldwide challenges should satisfy a maximum number of customers needs.

6. Use Cases

6.1 Use Cases Table

Function	Logged-in user	Not logged-in user
Authorization	-	+
Recovering password	-	+
Log out	+	-
Registration	-	+

User story	Me, as not authorised, but registered used should have an opportunity to start using application.
Business Rules	<ul style="list-style-type: none"> • User install app • User open app and see login form • User is able to enter login and password and to be logged in system - If user forgot his password he is able to recover - If user provide correct credentials then he will be logged in system as a player - User is able to choose game mode (single / multiplayer / free game without timer) - User is able to check world rating table - User is able to change his profile data • User is able to register his account by providing credentials
Validation	<ul style="list-style-type: none"> • All fields : required — please fill all fields to be able to login • Email: user exists — User with the provided email is already registered • Email: wrong format — Please, make changes your email to be in correct format • Password: wrong password — Please provide the correct password • Password: wrong format — Please, make changes your password to be in correct format