

CS442 Group Project Phase 4: Final Report & Demo

Team Number: 2

Project Topic: Sudoku

Phase Manager: Sandeep Singh Sandhu

Team Members: Hao Mei, Shruthi Adamala, Shorabh Dhandharia

Features List:

1. Gorgeous graphics and beautiful animations to make the game look more engaging.

We have used couple of android features such as custom views, swipe fragment, list view, dialogs, etc. to build the UI look attractive and user-friendly.

2. Detailed tutorial for beginners

The tutorial helps novice users in understanding the game and how to play it in a user-friendly way by sliding a couple of images which explains step-wise.

3. Code to generate puzzles of 3 x 3 grid size

We have used custom view to draw 3 x 3 puzzles.

4. Storing and Retrieving puzzles from the database

We have 30 puzzles for each level of difficulty loaded in the database which will be retrieved automatically when user installs the game.

5. Adjustability of Screen size

The game can be played in both portrait and landscape mode with ease and user friendly environment. This feature takes care of adjustment of screen size.

6. Background Music and Timer can be switched On/Off

Music

The music starts playing when the game is started, on disabling it in the settings will stop the music.

Timer

The timer can be disabled in the settings and then it disappears from the puzzle but it still running in the backend.

7. Timer to check the time taken for a puzzle

The timer is displayed in the puzzle indicating the user the time taken to complete the game. It also helps to calculate the high scores.

8. Error-Checking to ensure proper gameplay

When the user enters a number which is already present in the sector, row or column it will display in red indicating that particular number is not applicable in the block. If the user types in the correct number then it is displayed in default color that is black.

9. Edit Feature to track and help solving difficult puzzles

When user clicks on edit mode image button in the puzzle, user can put notes in the form of numbers in each cell of the puzzle to keep track of moves. As user disables the edit mode image button in the puzzle, user can put only single digit numbers in each cell of the puzzle. The user can identify if the edit image button is enabled or disabled looking at the image button. If it is enabled the color of the button changes to blue otherwise it is in disabled state.

10. Save the state of the game once the activity is stopped

The entire puzzle's state along with the timer is saved on pause and on stop activity.

11. Undo, Restart & Clear all notes

Undo

This option is disabled initially when the game starts as there are no changes done. Once the user does changes the option gets enabled and then the user can select it if any previous changes are to be known to the user.

Restarts

This option restarts the game, displaying the game without any chances done previously. Thus starting the game from the scratch.

Clear all notes

This option clears all the notes which have been noted earlier to keep track of the game.

12. Provide an upgrade link to add more puzzles on completion

We have added an upgrade button which will add 10 more puzzles in each level of difficulty and display a toast message after the puzzles are loaded.

13. Add a Resume button on the Main Activity

We have included a resume button on the main activity which will load and resume the last playing puzzle.

14. High Score Records

Based on the time taken by the user in each of the difficulty level, we are displaying the least time taken by a user to play a particular puzzle in a difficulty level as the highest score.

Class List:

The project can be broadly classified into three categories, which are as follow:

- **Game Logic**
- **UI Design and Logic**
- **Database Logic**

Game Logic

The game logic has the classes which takes care of the different aspects of game as the Sudoku cell block, Sudoku game engine, cell collection etc. The complete list is as follow:

- **Command Classes:** These classes are used to store the values in the stack which is used to execute the functionalities such as the clear, edit, undo set cell value etc. The list of classes which comprise the commands group is as follow:
 1. AbstractCommand
 2. ClearAllNotesCommand
 3. CommandStack
 4. EditCellNoteCommand
 5. SetCellValueCommand
- **Cell Classes:** The logic and the game generator is handled by these classes. The difficulty level , grid size and each cell along with the numbers displayed are being generated in the classes which are as follow:
 1. Cell
 2. CellCollection
 3. CellGroup
 4. CellNote
 5. SudokuGame

Database Classes: These classes are used to store, retrieve and update the puzzles onto the device's storage. It also keeps a tab of all the puzzles which have been loaded into the game. The database used for this purpose is SQL-lite and the classes are as follow:

- DatabaseHelper
- SudokuDatabase

User Interface Design Classes:

The User Interface Design Classes as reflected from the name takes care of the User Interface of the game. It has classes which take care of the functionalities like the game tutorial, grid layout and design, the main navigation menu, etc. The complete list of the classes is as follow:

- About
- GameTimeFormat
- Music
- Numpad
- Prefs
- SplashScreen
- Sudoku
- SudokuBoardActivity
- SudokuListActivity
- SudokuPlayActivity
- SudokuRecords
- Timer
- TutorialActivity

The minimum API level supported by the app is 11 and the targeted API level for the app is 23.

Test Cases:

Background Music and Keypad sound can be switched On/Off

- Test Case 1: On Start Background Music is On →Passed
- Test Case 2: User goes to settings and unchecks the Music, background music is switch off. →Passed
- Test Case 3: User goes to settings and checks the Music, background music is switch on. →Passed

Detailed tutorial for beginners

- Test Case: User clicks on Tutorial Button on the Main Activity. A new activity is opened which displays an image slider. →Passed

Adjustability of Screen Orientation & Scalability

- Test Case 1: All the activities when rotated to landscape mode displays the content as it is shown in Portrait mode. →Failed
- Test Case 2: All the activities when rotated to landscape mode displays the content as it is shown in Portrait mode. →Passed
- Test Case 3: The layout Sudoku Play Activity is modified for landscape mode to fit the Sudoku board and numpad according to screen size. →Passed

Save the state of the game once the activity is stopped

- Test Case 1: User clicks the back button while playing the game. The state of the game and timer is saved. →Passed
- Test Case 2: User comes back to the same game, the game and timer is restored back from where she left. →Passed

Storing and Retrieving puzzles from the database

- Test Case 1: There are 30 puzzles for each level of difficulty which is loaded when the user selects the appropriate difficulty level from the main activity. →Passed
- Test Case 2: The Puzzles state is saved and restored in case user quits the application and comes back. →Passed

Code to generate Puzzle of 3 x 3 grid size

- Test Case 1: A 3 x 3 grid is designed using custom view which is displayed in the game activity. →Passed
- Test Case 2: The grid is also displayed in the Sudoku list activity for designing purpose. →Passed

Timer to check the time taken for a puzzle

- Test Case 1: If timer is turned on in the settings, a timer is displayed on the title bar of the Sudoku game activity in the format of MM:SS. →Passed
- Test Case 2: If timer is turned off in the settings, the timer is hidden from the title bar of the Sudoku game activity but the timer is running in backend. →Passed

Edit Feature to track and help solving difficult puzzles

- Test Case 1: When user clicks on edit mode image button in the puzzle, user can put notes in the form of numbers in each cell of the puzzle to keep track of moves. →Passed
- Test Case 2: When user disables the edit mode image button in the puzzle, user can put only single digit numbers in each cell of the puzzle. →Passed

Error-Checking to ensure proper gameplay

- Test Case 1: According to game rules, if user enters a number which is already present in the sector, row or column it will display in red. →Passed
- Test Case 2: If user enters a number which is not present in the sector, row or column it will display in black which is default colour. →Passed

Undo, Restart & Clear all notes

- Test Case 1: When user clicks undo, the last move is deleted →Passed
- Test Case 2: When user clicks on restart, a dialog box appears to ask for its confirmation and after positive response the puzzle is restarted and timer is reset→Passed
- Test Case 3: When user clicks on clear all notes, a dialog box appears to ask for its confirmation and after positive response all the notes in the puzzles has been cleared→Passed

Add a Resume button on the Main Activity

- Test Case 1: When user clicks Resume Game, the last playing puzzle is loaded from the state where it was stopped/paused. →Passed
- Test Case 2: After installing the game if user clicks Resume Game directly without even starting a game → Failed
- Test Case 3: After installing the game if user clicks Resume Game directly without even starting a game, a toast will display saying no game to resume → Passed

Record High Scores

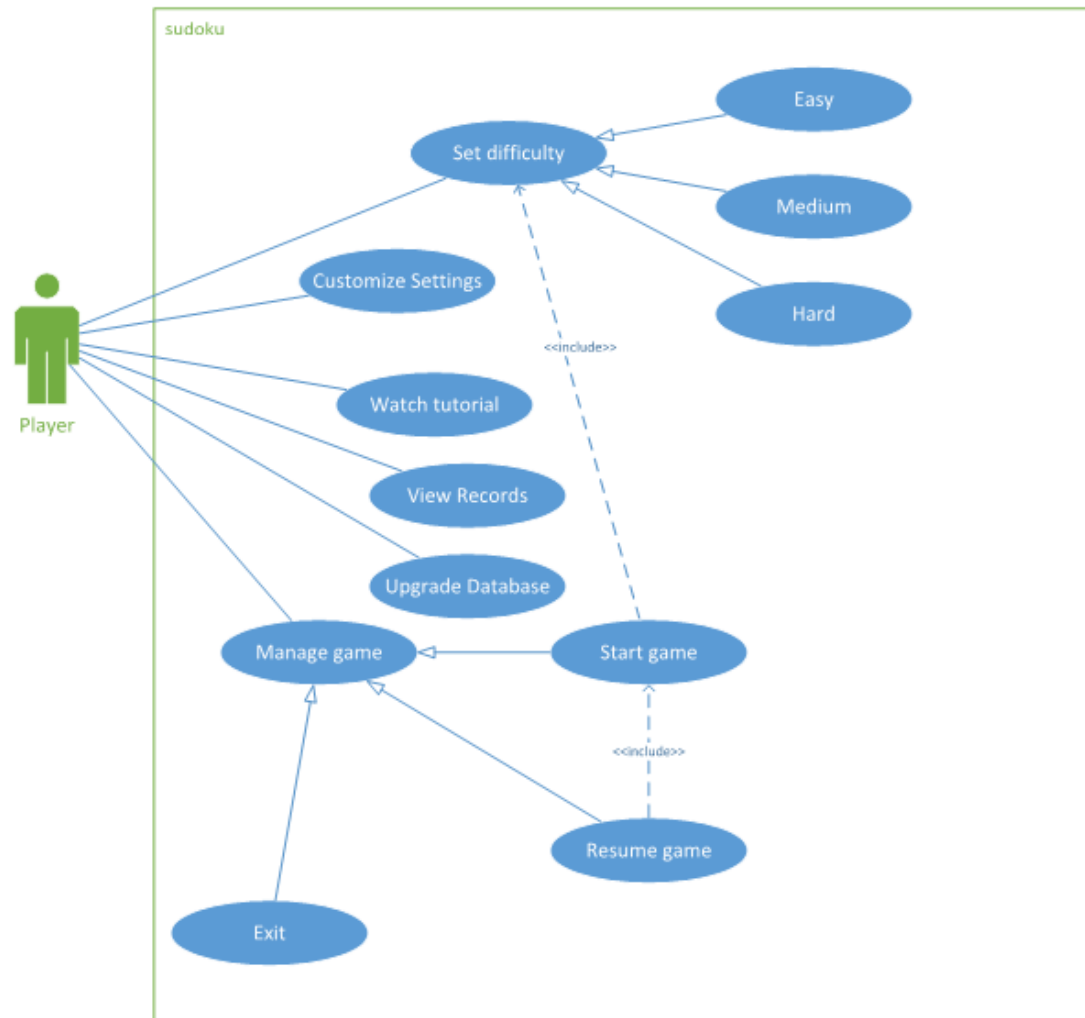
- Test Case: After completion of game, game time is captured and displayed in high scores if it beats the high score → Passed

Provide an upgrade link to add more puzzles on completion

- Test Case 1: When user clicks on upgrade, 30 new puzzles are loaded and a toast message is displayed. →Passed
- Test Case 2: If user clicks on upgrade again. →Failed
- Test Case 3: If user clicks on upgrade again, it will display a toast saying it has been loaded already. →Passed

Diagrams:

Use Case Diagram:

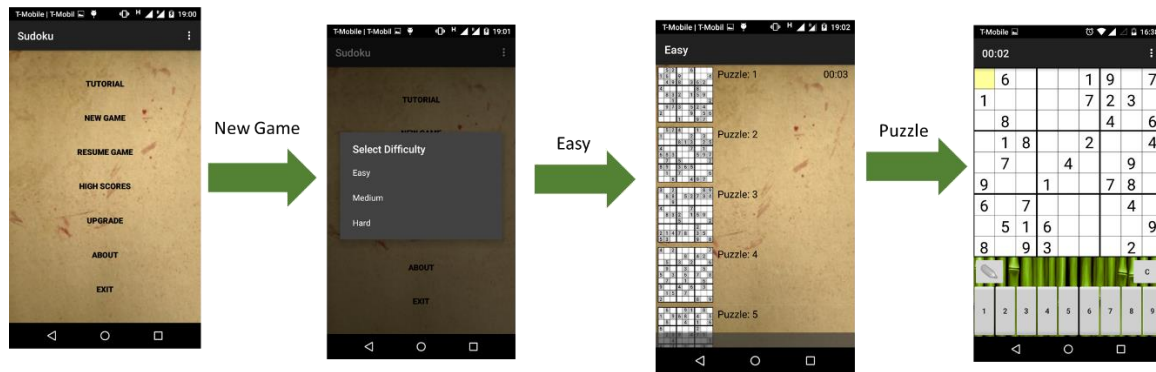


Use Case Scenarios:

After installation of the game, user launches the game:

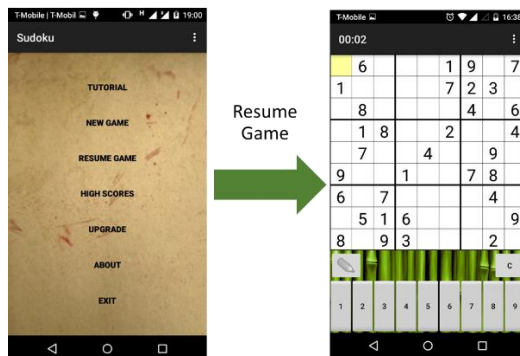
Scenario 1:

1. User sees the main activity screen with couple of options such as resume game, new game, tutorial, about, high scores, and exit.
2. User clicks on new game, chooses easy from the dialog box and goes to Sudoku list.
3. User then clicks on one of the puzzles and starts playing.



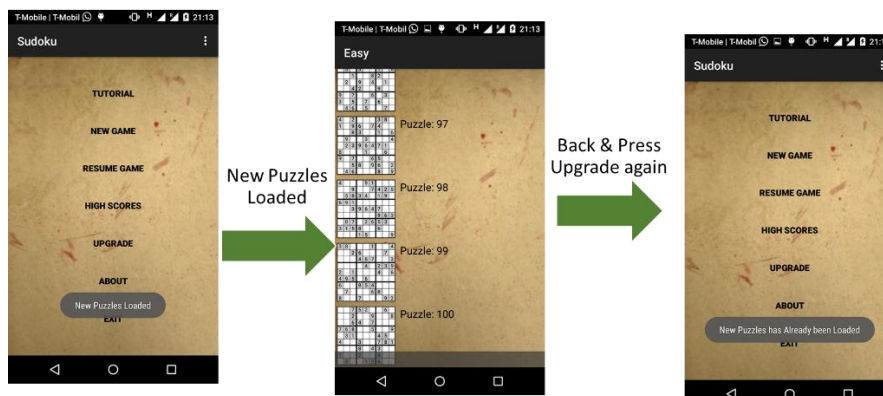
Scenario 2:

1. User relaunches the game
2. User sees the main activity screen with couple of options such as resume game, new game, tutorial, about, high scores, and exit.
3. User clicks on resume button, game starts from where user has left.

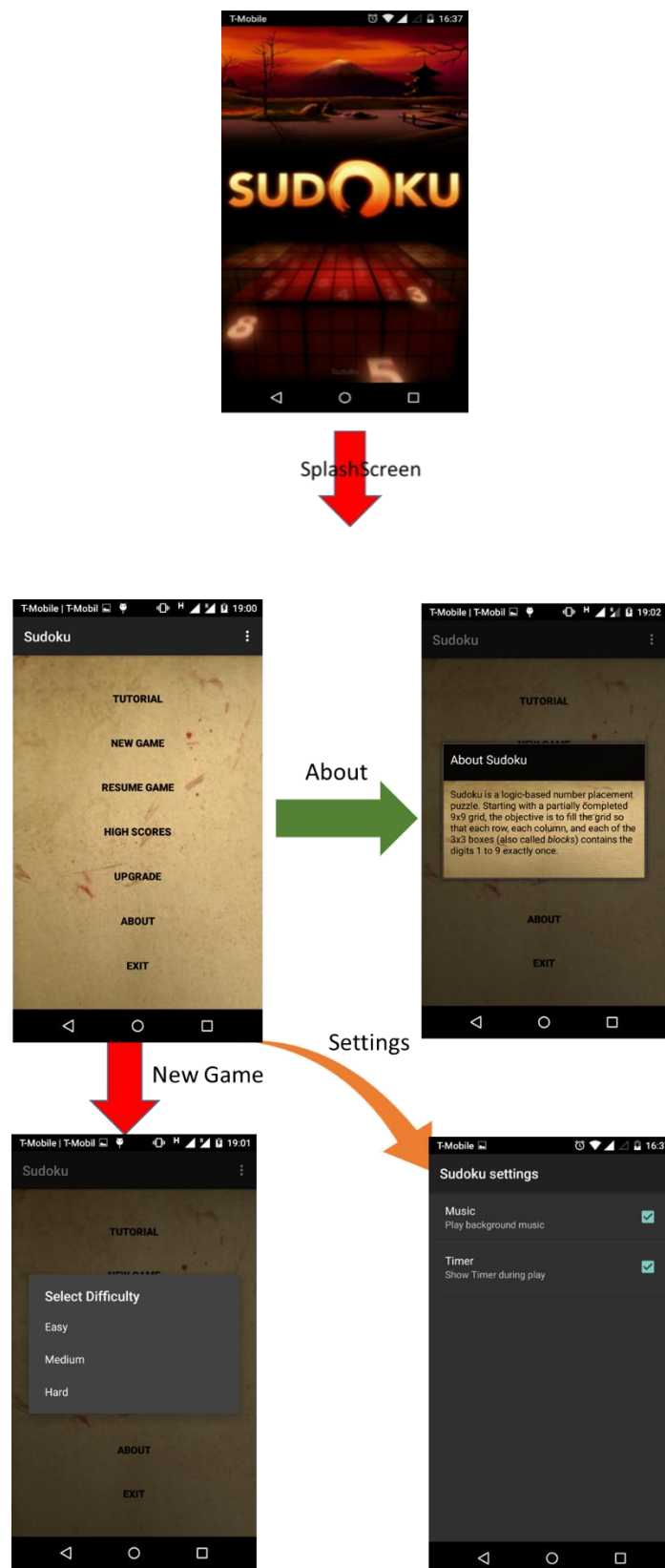


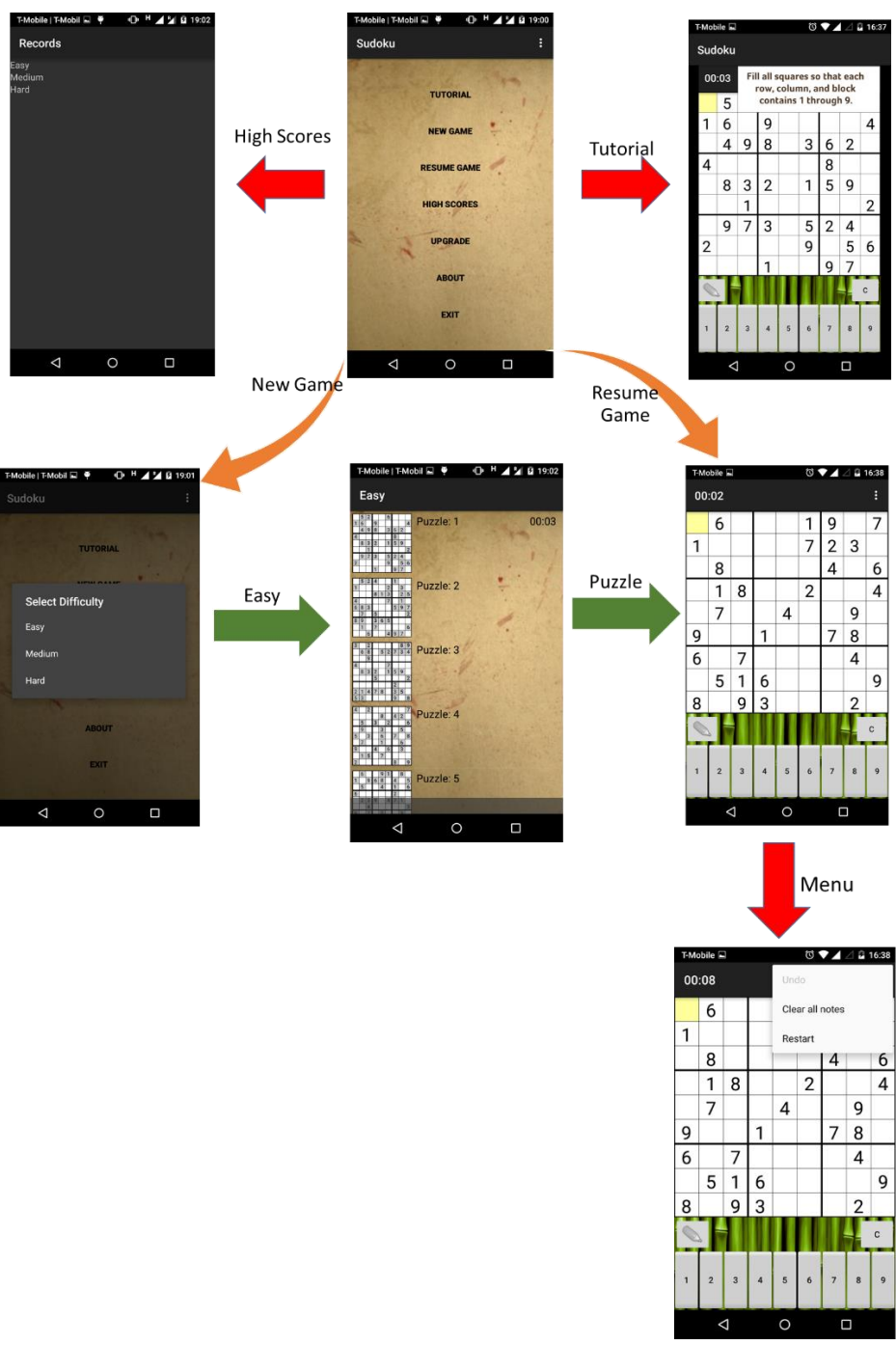
Scenario 3:

1. User sees the main activity screen with couple of options such as resume game, new game, tutorial, about, high scores, and exit.
2. User clicks on upgrade, a toast message stating the puzzles has been loaded appears.
3. User then clicks on new button to check for the new puzzles in each of the difficulty levels.
4. User clicks on upgrade again, a toast message appears saying the puzzles has already been loaded.



UI Flow:

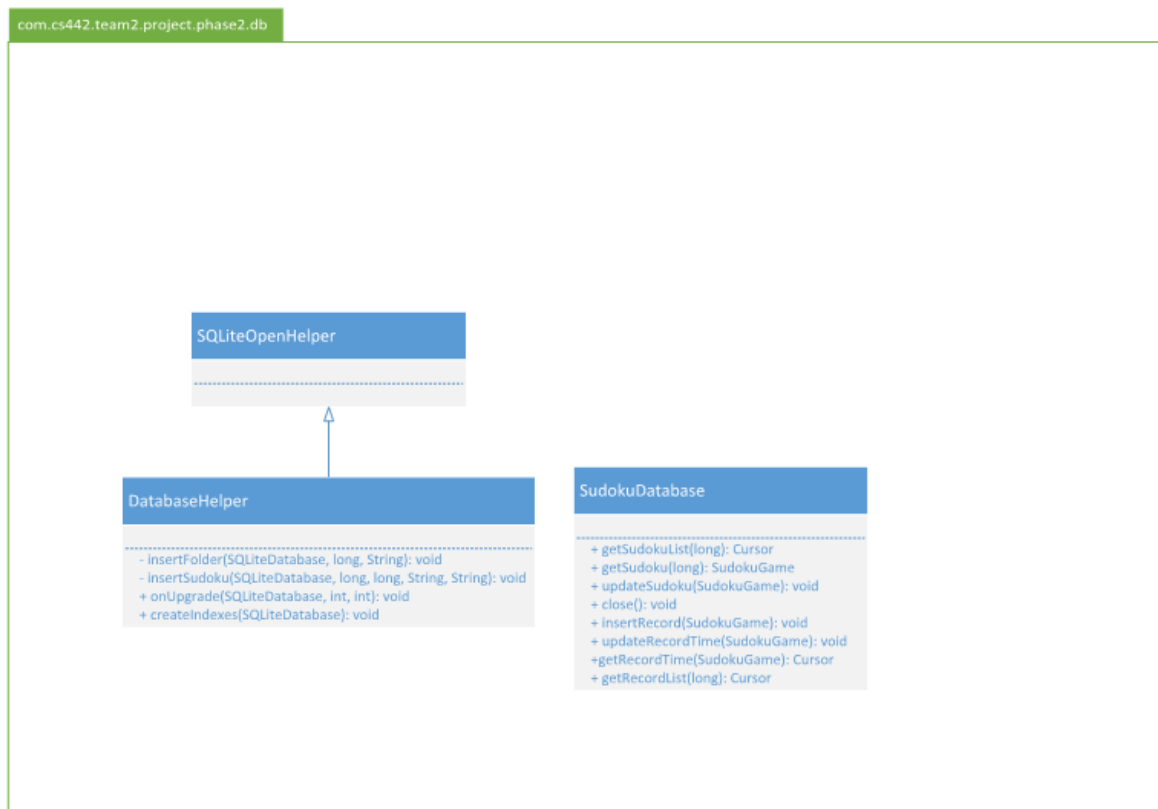




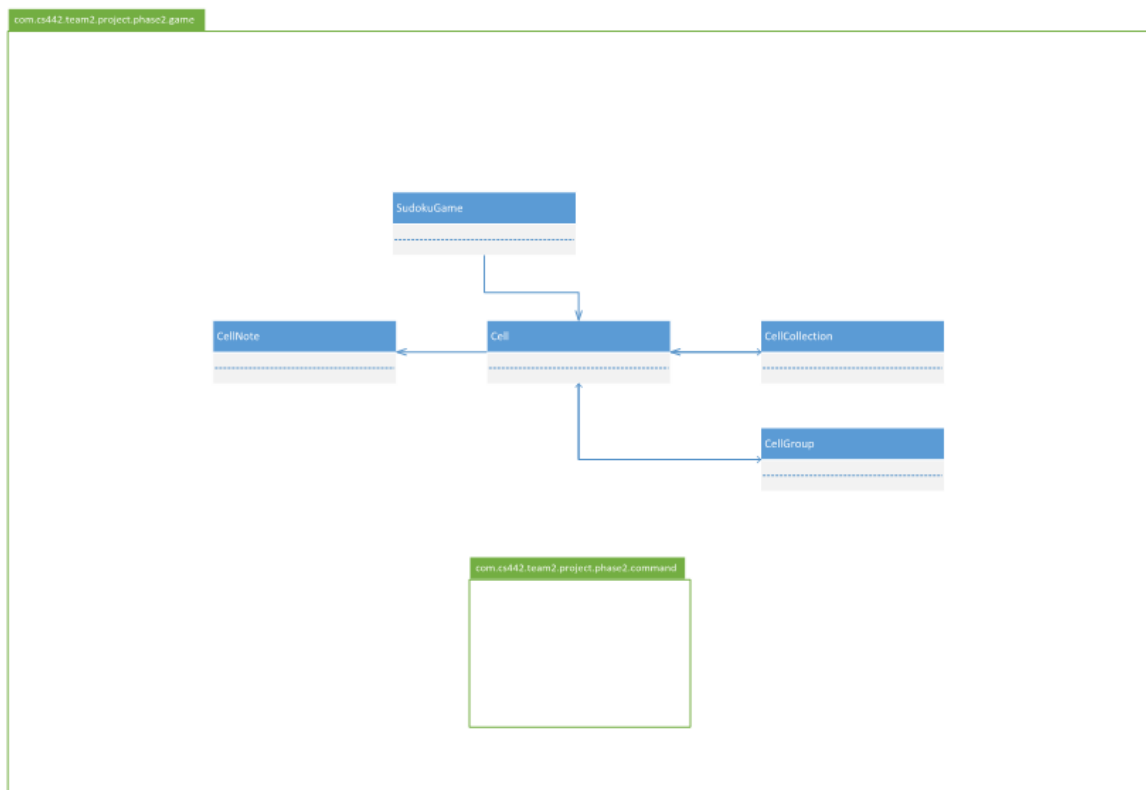
Class Diagrams:

Since the class diagrams are too huge to fit in this document we have break into 3 different class diagrams based on the packages. To view clear version of class diagram please refer the Visio files uploaded along with the pdf.

Database package



Game Package



Main Package

