

Criterion B: Design

Table of Contents

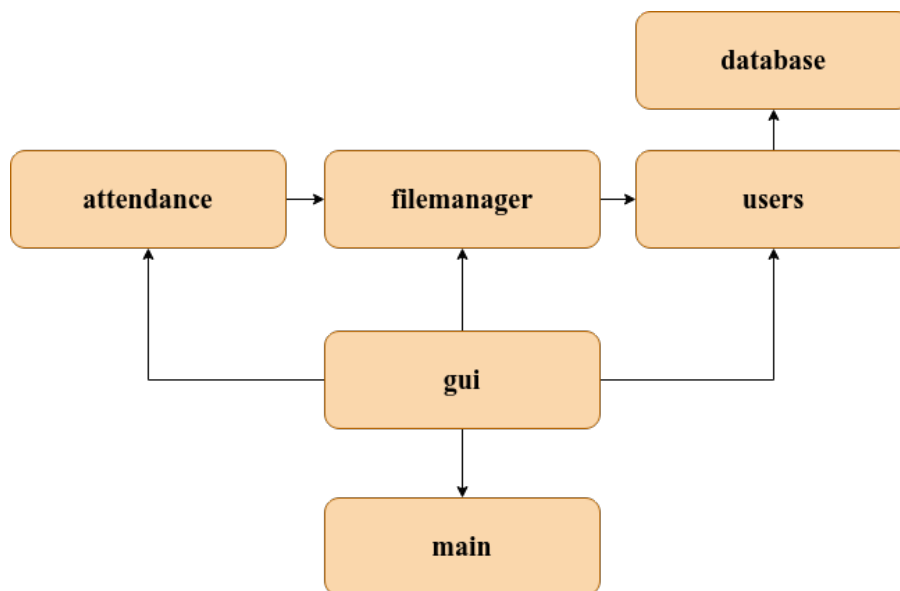
Modules and Their Relationship.....	1
UI Classes and Their Relationship.....	2
UML Diagrams.....	3
Process Flowcharts.....	4
Database Tables and Fields.....	15
GUI Design.....	16
Test Plan.....	21

Modules and Their Relationship

The following are the modules the program will be divided into and their functions:

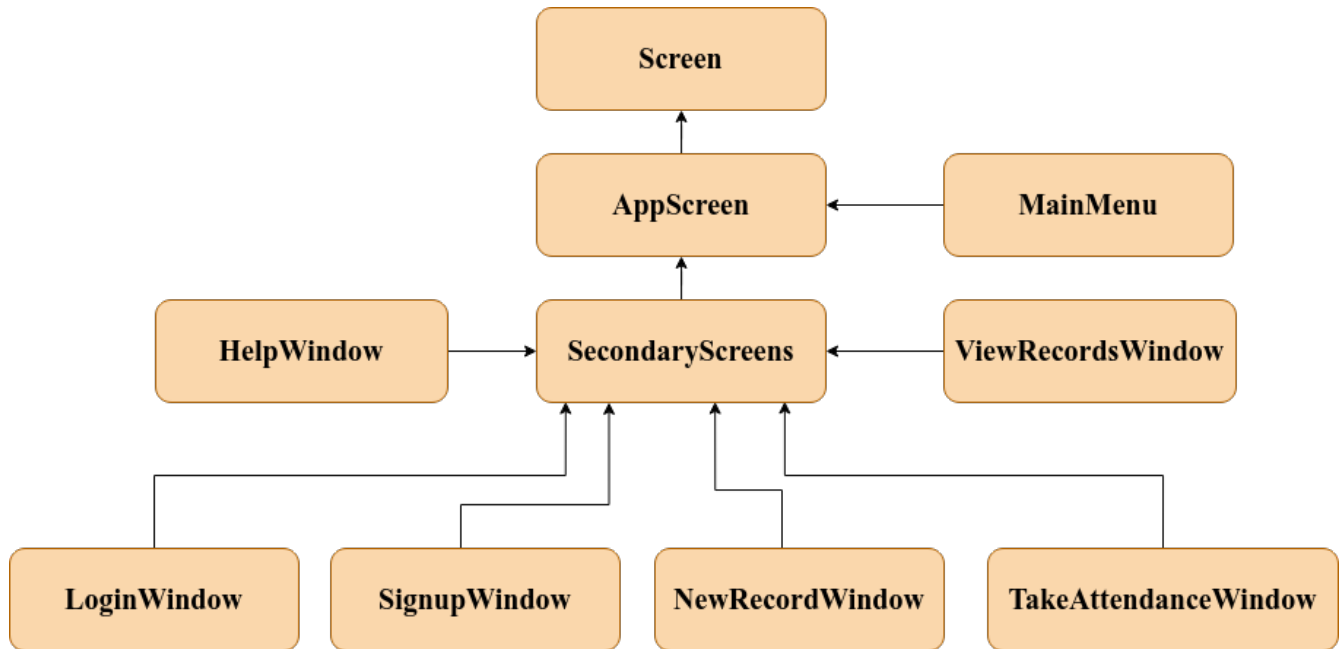
1. **main:** Builds the GUI Kivy file.
2. **gui:** Written in the Kivy language unlike the other modules, includes all of the code related to the design of GUI as well user interaction with it is defined here.
3. **filemanager:** Handles all the functions related to the spreadsheet file to be generated and update records in.
4. **attendance:** Handles taking new attendance records or updating previously existing attendance records.
5. **users:** Handles creating new user and setting current user and authorizing user credentials
6. **database:** Handles all things related to database such as adding new user to database, getting user information from database, hashing passwords and comparing them to the ones already in the database, etc.

The diagram presented below shows the relationship between the various modules. Here the arrows suggest which module has imported classes of which other module. For instance, the module 'users' imports some classes from the 'database' module.



UI Classes and Their Relationship

This section shows the relationships between the different classes used for UI design. It is important to note that Kivy treats every widget as a class. Hence, all UI elements such as buttons, screens, text fields, etc. are classes. Only the relationship between the different screen classes have been shown below as it not only represents what and how many screens exists in the application but also gives information about program flow.



UML Diagrams

The following illustrates the main classes used in the program excluding the ‘main’ and ‘gui’ modules.

<i>User</i>
<i>current_user : NoneType</i>
<i>auth_user(username, password)</i> <i>init_new_user(username, password, confirm_password)</i> <i>new_user_dir(username)</i> <i>set_current_user(username)</i>

<i>AttendanceHandler</i>
<i>chat_file</i> <i>excel_file</i> <i>excel_names : list</i> <i>excel_sheet</i>
<i>get_names_excel()</i> <i>get_present_students()</i>

<i>FileManager</i>
<i>create_excel_file(names, name_of_record)</i> <i>delete_record(instance)</i> <i>find_file(name_of_record, path)</i> <i>get_recent_searches()</i> <i>open_file()</i> <i>process_names(names)</i> <i>return_record_names(records)</i> <i>save_recent_searches(search)</i> <i>start_file(file_path)</i> <i>update_excel_file(chat_file, excel_file)</i> <i>view_record(instance)</i>

<i>DbManager</i>
<i>add_user(username, password)</i> <i>check_db()</i> <i>check_password(password, hashed_pwd)</i> <i>confirm_password(password, confirm_password)</i> <i>connect_user_db()</i> <i>get_id()</i> <i>get_pwd(username)</i> <i>hash_password(password)</i> <i>init_user_db()</i> <i>username_exists(username)</i>

<i>Attendance</i>
<i>name_of_record</i>
<i>take_new(names_of_students)</i> <i>update(chat_file)</i>

Process Flowcharts

The flowcharts presented below illustrate how each major feature of the software will be implemented.

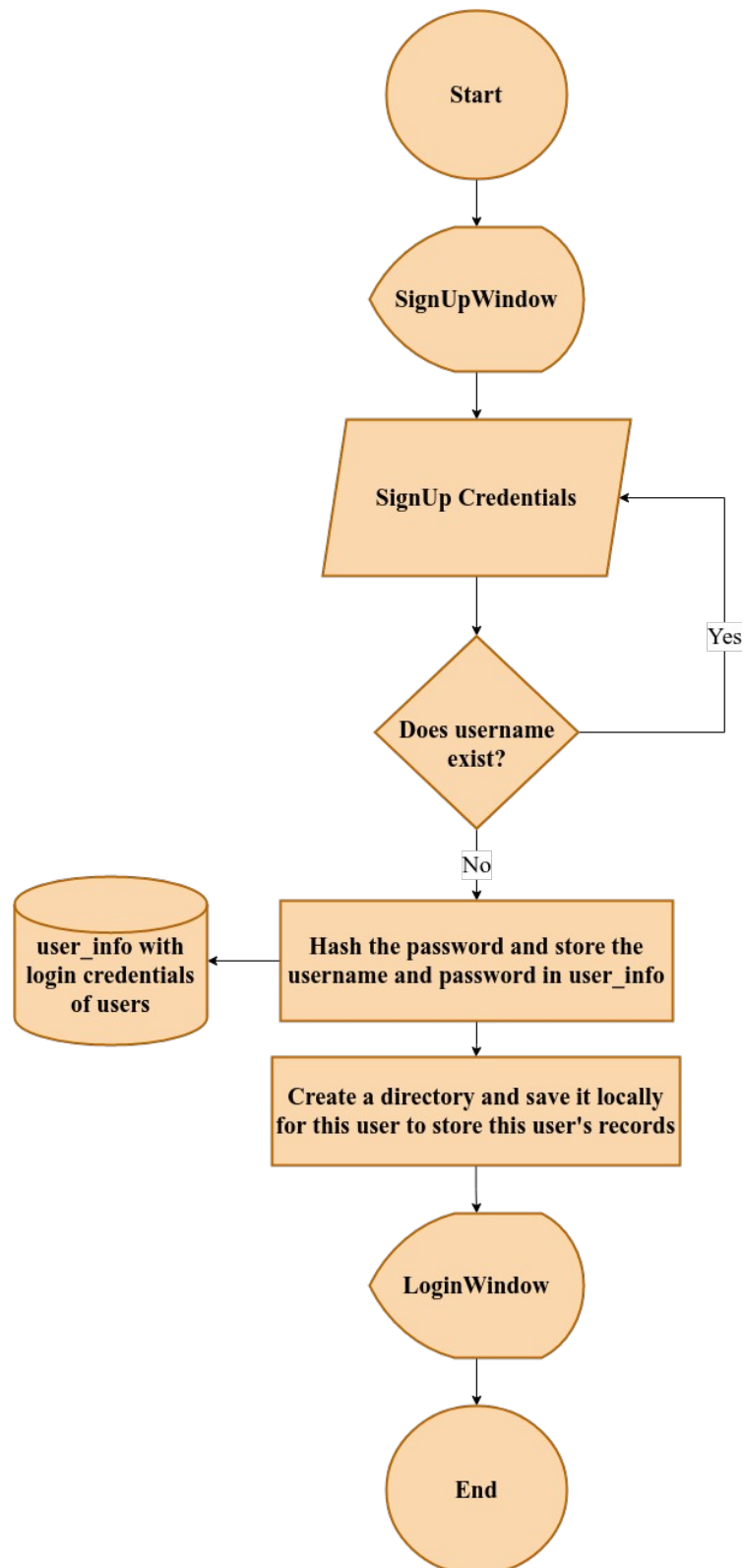


Figure 1: Flowchart for SignUp Feature

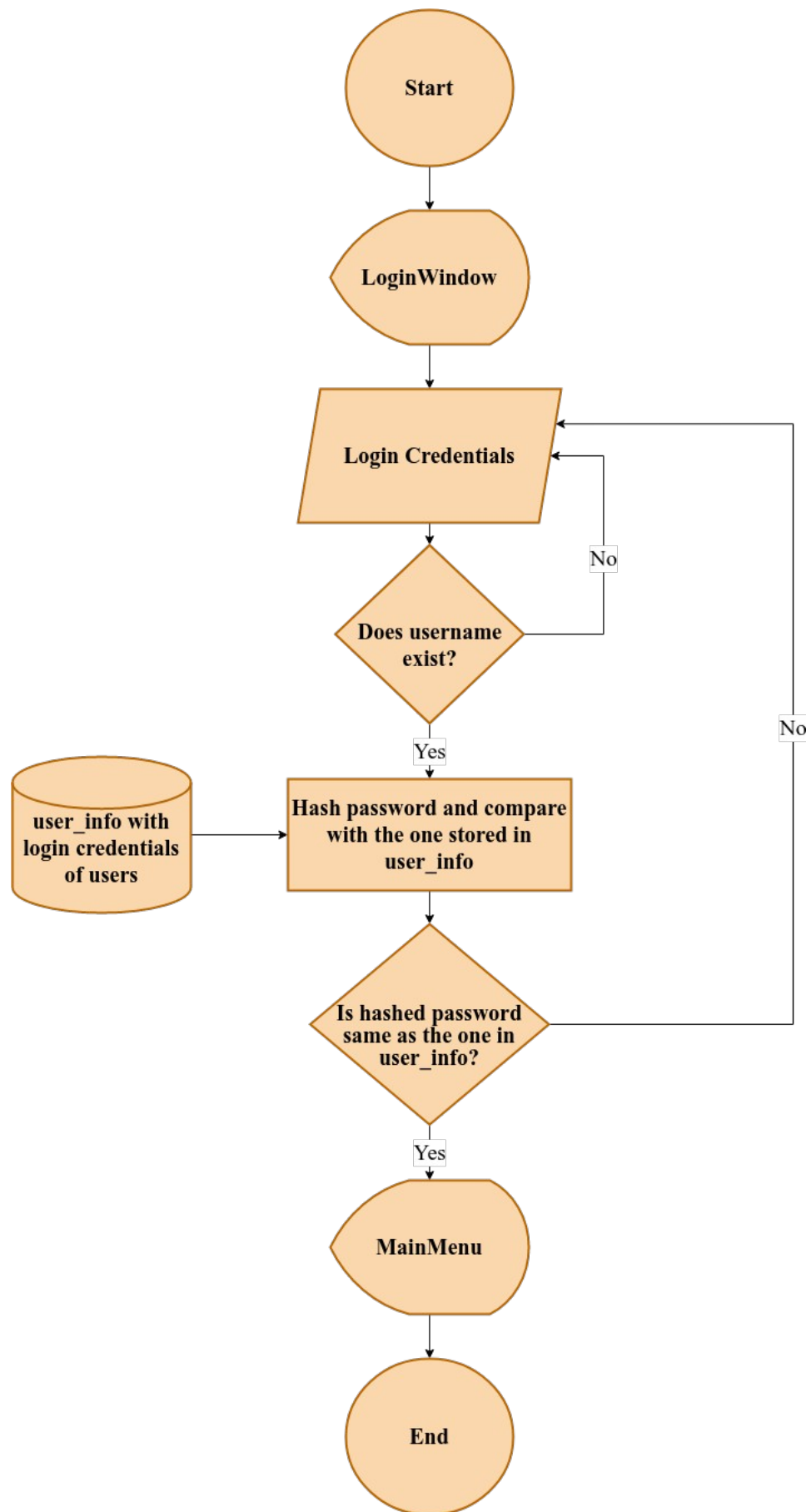


Figure 2: Flowchart for Login Feature

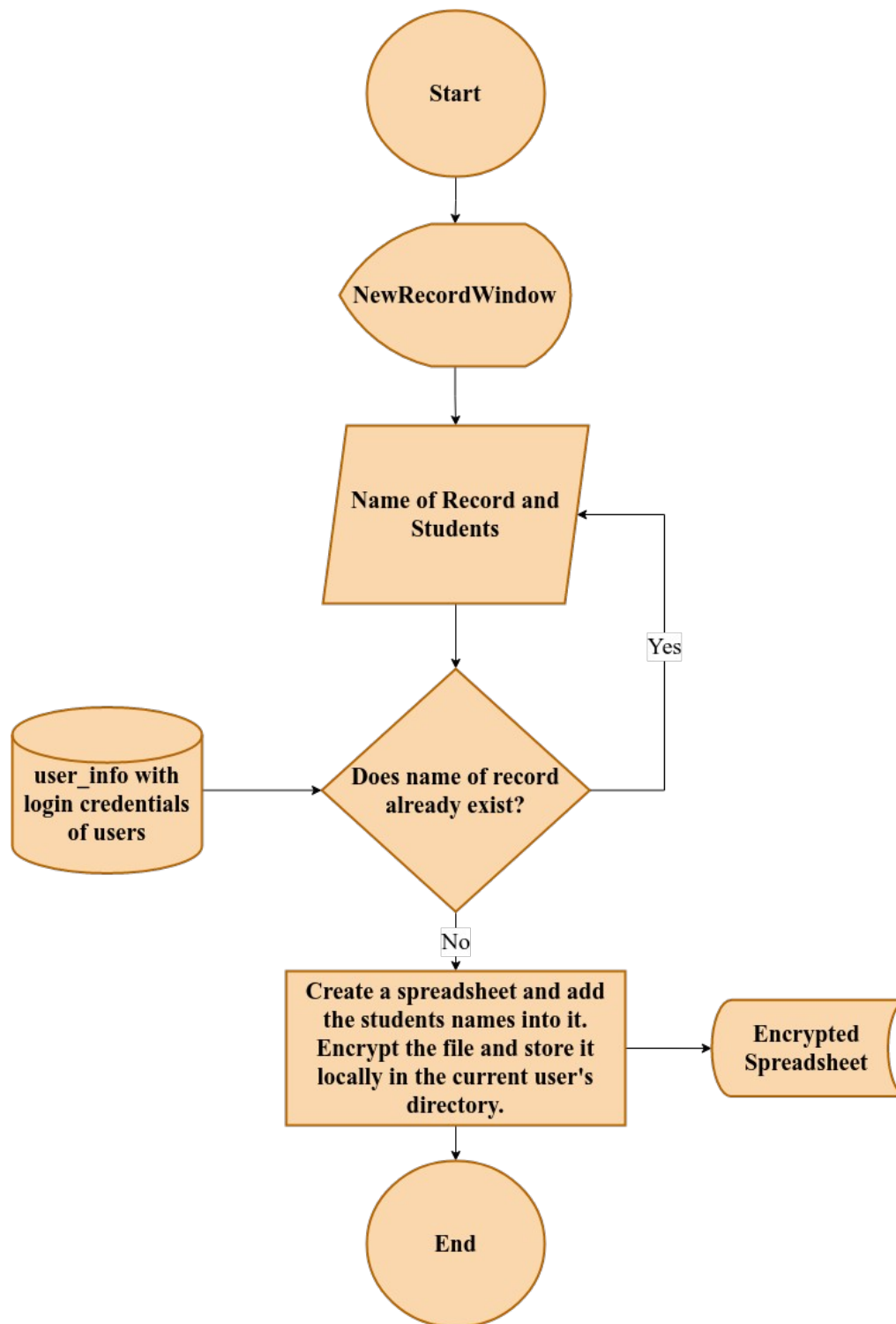


Figure 3: Flowchart for Add/Create a New Record Feature

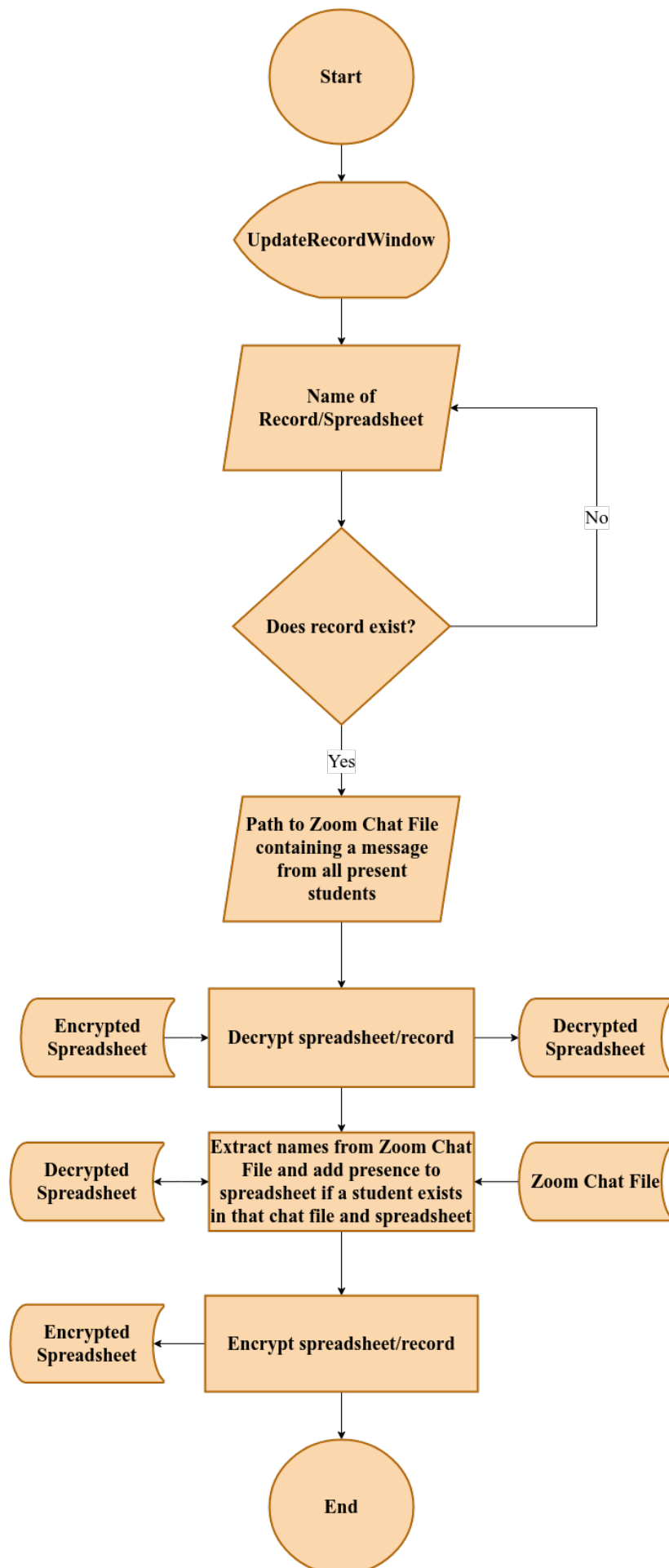


Figure 4: Flowchart for Update Records Feature

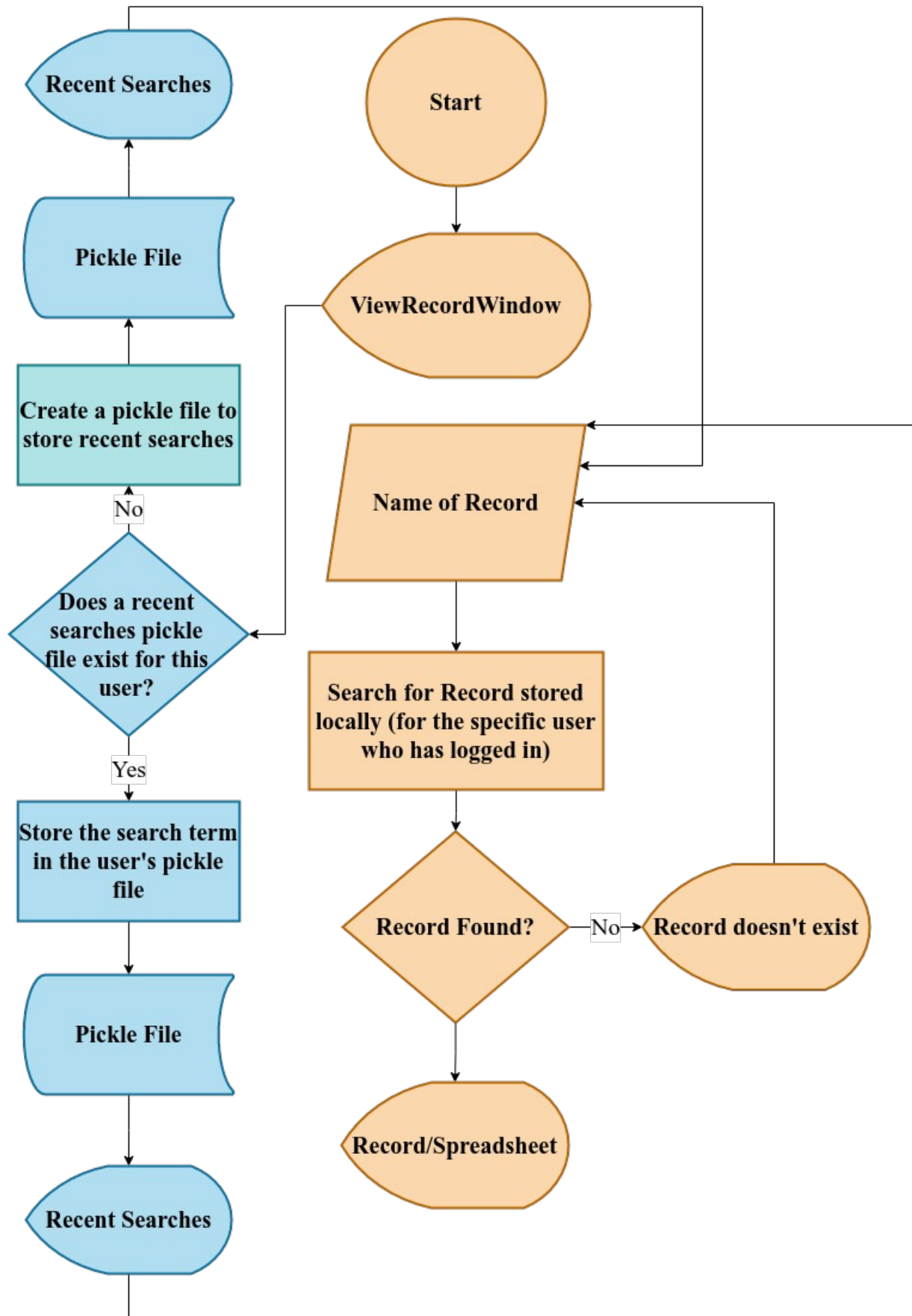


Figure 5: Flowchart for Searching Records (Orange) and Recent Searches (Blue) Features

Database Tables and Fields

Only a single database is required for this application. This database 'user_info' will be storing information about login credentials of users (their usernames and passwords). The database that will be used will be a SQLite database stored locally. The Data Dictionary is illustrated below:

user_info		
Field Name	Data Type	Description
SN	INTEGER	Stores the serial number
Username	TEXT	Stores the username of the user
Password	TEXT	Stores the hashed password of the user

GUI Design

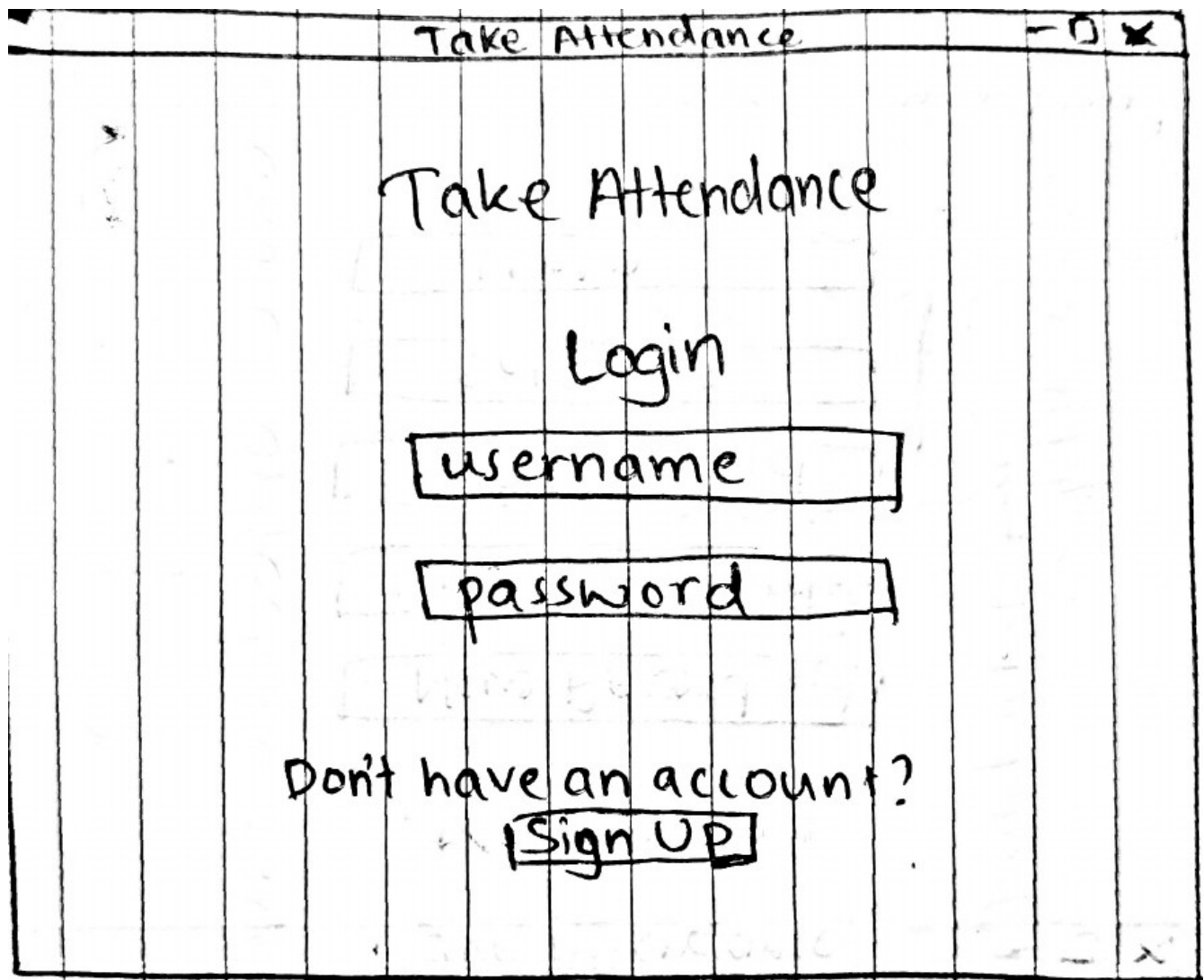


Figure 6: LoginSignupWindow Screen

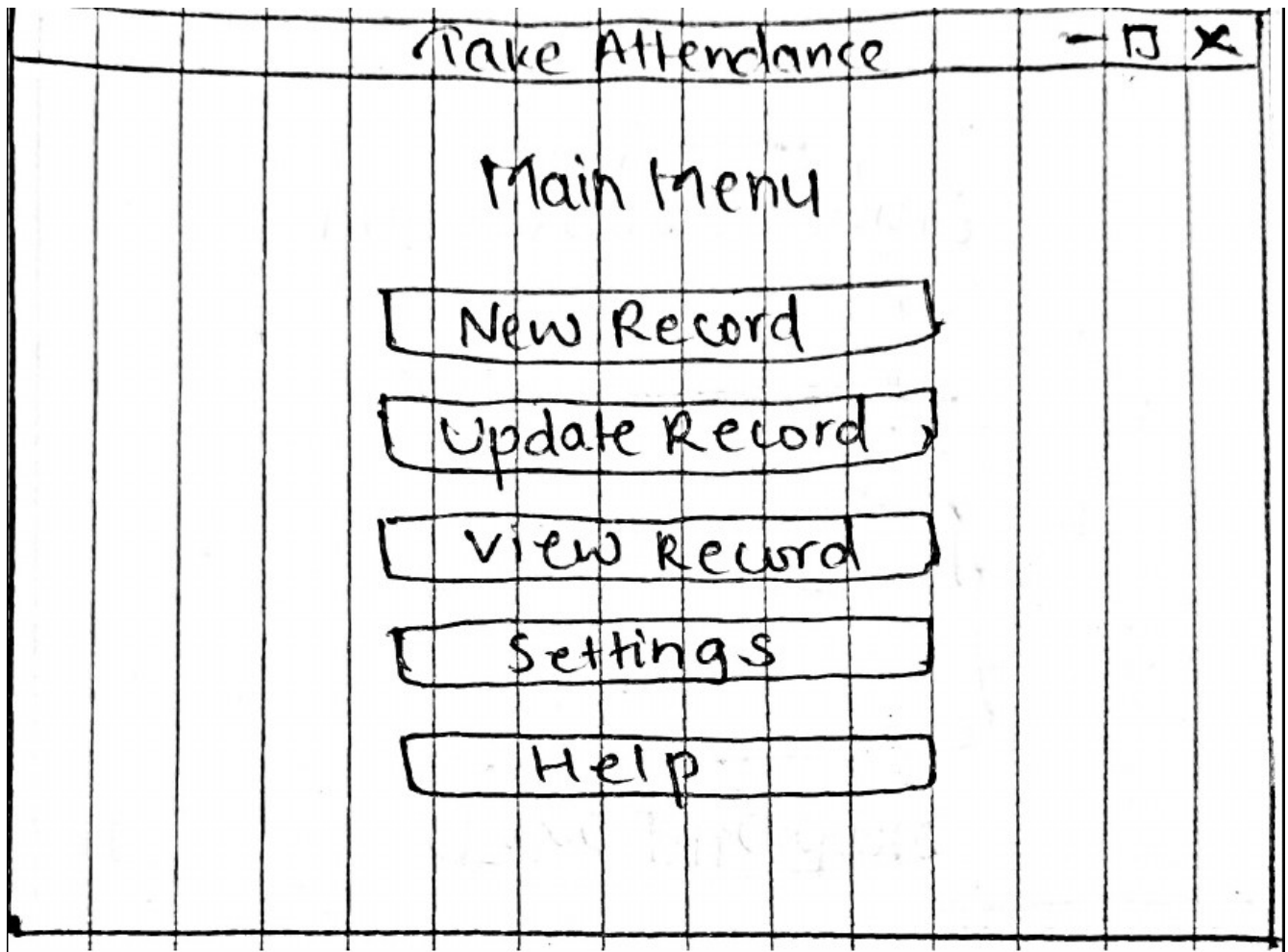


Figure 7: MainMenu Screen

Take Attendance										-	□	X
New Record												
Enter Name of Record												
<input type="text"/>												
Enter Name of Students												
<input type="text"/>												
<input type="button" value="Create Record"/>												
												<input type="button" value="Main Menu"/>

Figure 8: NewRecordWindow Screen

Take Attendance

Update Record

Enter Name of Record

Update Record

main menu

Figure 9: UpdateRecordWindow Screen

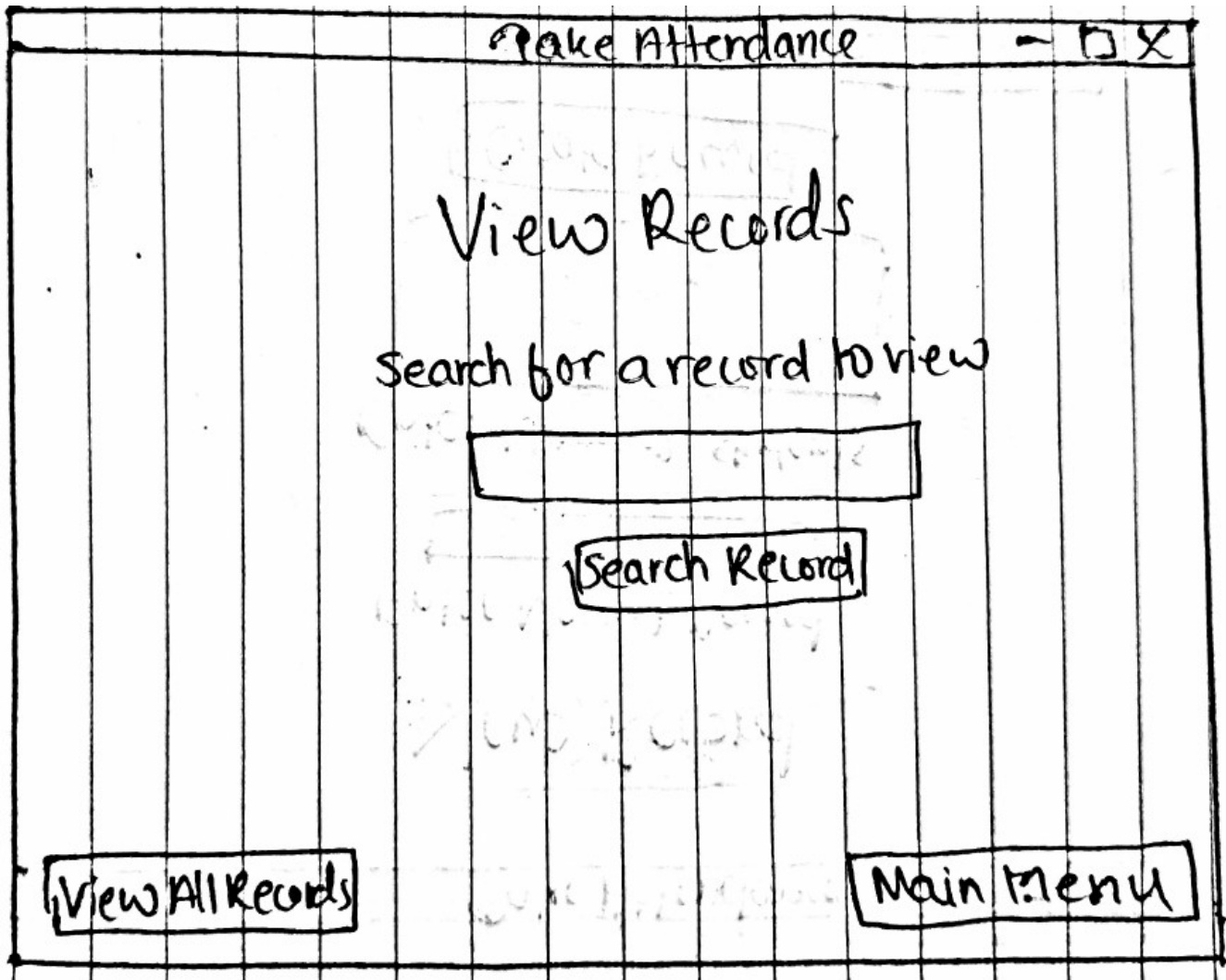


Figure 10: ViewRecordsWindow Screen

Test Plan

Action To Test	Success Criteria Tested	Method
How user-friendly the GUI is.	1	The attendance can be taken with least amount of click possible, the typography is appropriate, there are suggestions on screen so that the user knows what to input.
Signing Up And Login	2, 3, 4, 5	Create a new user account using the Sign Up feature and check if the credentials have been saved correctly in the database. Then, try to login with the new user accounts created and check if successfully login is possible.
Creating New Records	6, 7	Create a new record using the software. Here, a spreadsheet should be generated by the program. Also, check if the Name of students have been properly sorted in the spreadsheet. Finally, check if the password protection of records works.
Updating a Record	8, 9	Update (add/remove names and add attendance) the record previously created while testing the create new records function. Check if the spreadsheet is properly formatted and the attendance has been accurately updated.
Deleting a Record	8	Delete the record previously created while testing create new and update record functionalities. Check if the spreadsheet file has been completely deleted.
Search and View Records	8, 10	Create a new record and search for them. Check if the search results were correct. Try to view the record searched for. See if the record asked to view is the correct one and a spreadsheet is

		opened.
Check for conflict in record names between users	5, 10	Create 3 new users and create a record (each with the same name) from the 3 accounts. Use the search functionality to see if searching for the created record displays all 3 records or just the record created by the particular user.
Display alert and error messages	1, 12	For every place where the user is able to input value check if the program displays the correct message for normal data, boundary data and abnormal data.