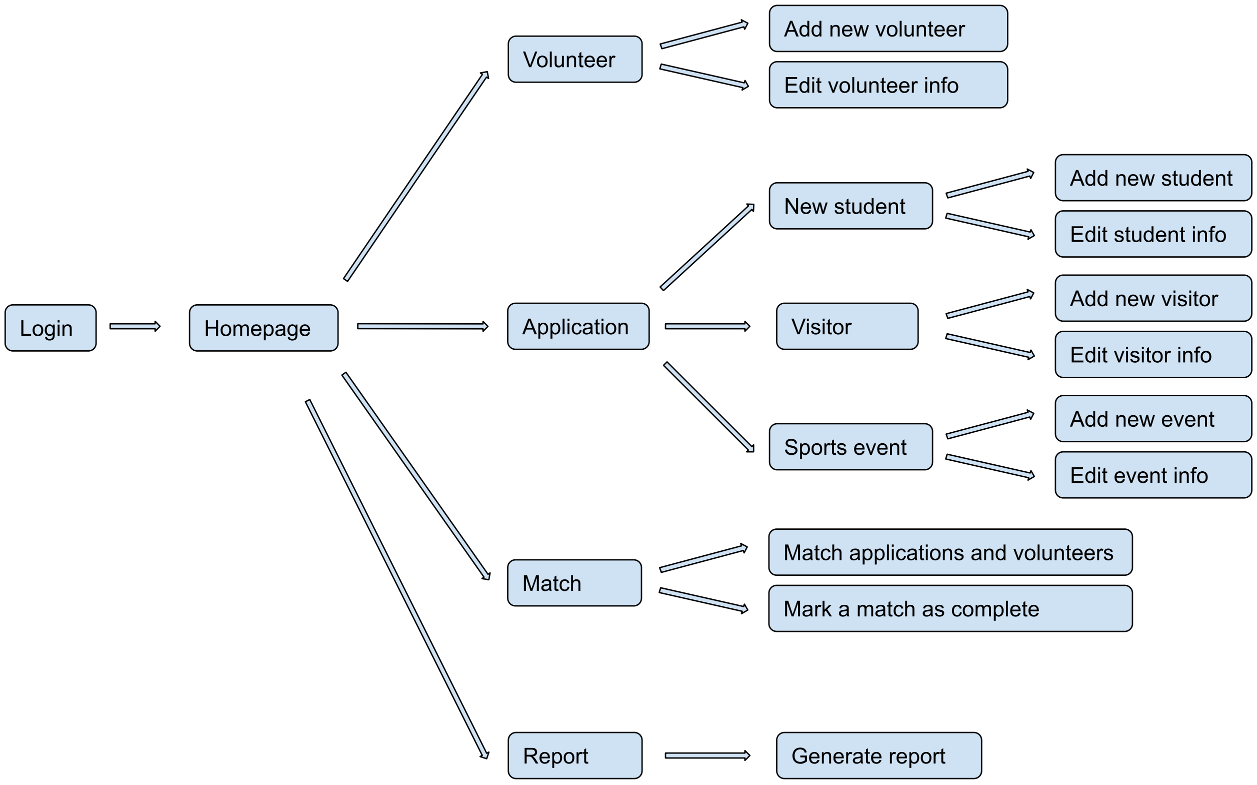
Criterion B: Design

Modular design:

The diagram below showcases all the modules that I will create in my program. Each module is designed to perform a particular task and is linked to the homepage.



Proposed tables:

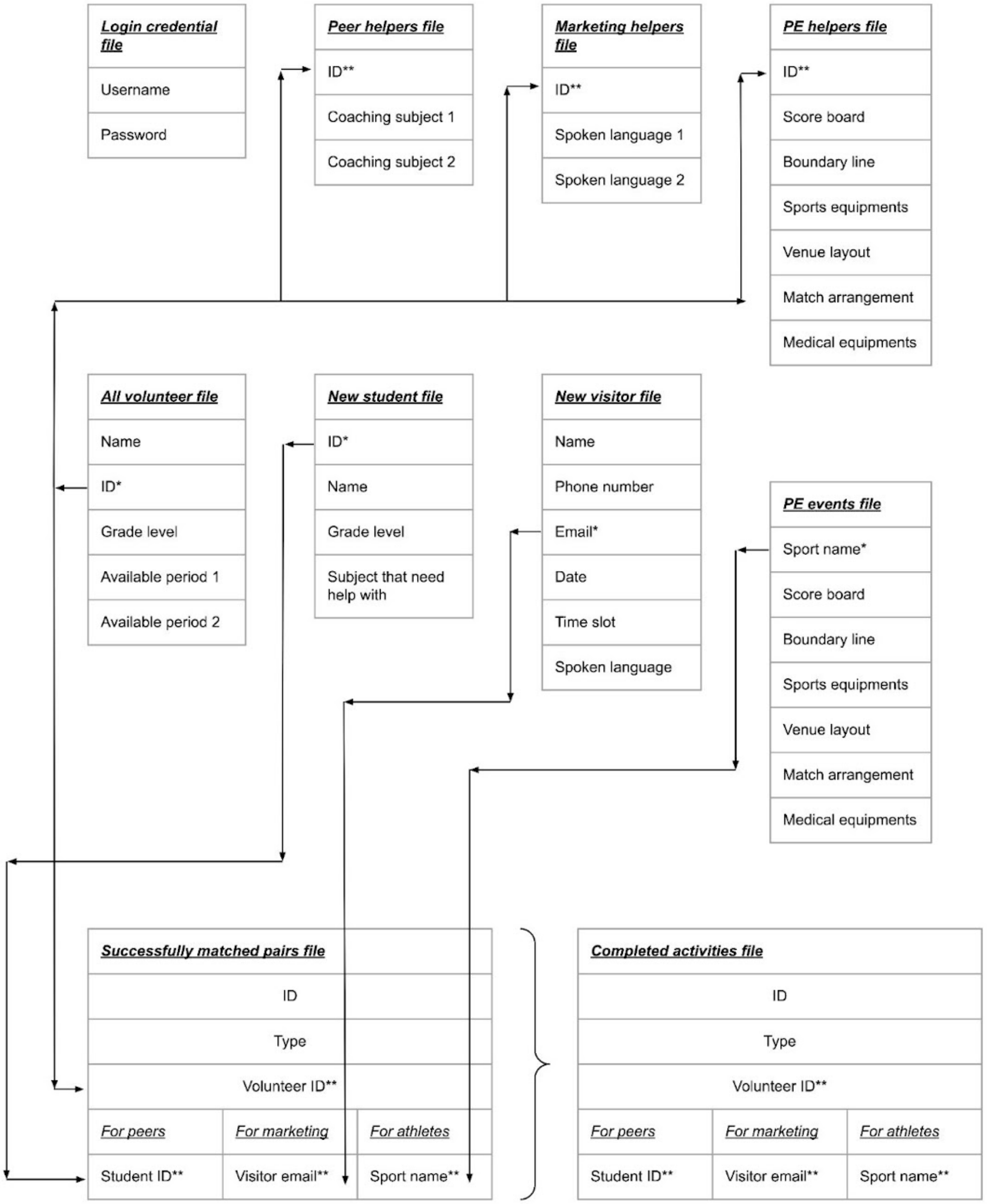
Table name	Details
Login → login information	This table identifies the user by the inputted login credentials and will allow the user to enter the system only if they match with the recorded username and password. Reads file: ● Login.txt
Volunteer → Add new volunteer	This table saves the inputted information of a volunteer, whereby the general information will be stored in a file and depending on the type of volunteer chosen, corresponding information will be stored in the related files. Creates files: ● Volunteer.txt (for general information) ● Peer.txt (for peer helpers) ● Marketing.txt (for marketing helpers) ● PE.txt (for PE helpers)
Volunteer → Edit volunteer information	This table reads the information of a volunteer based on the ID chosen, and a button could be clicked to save the changed information of the volunteer. Reads and writes files: ● Volunteer.txt ● Peer.txt ● Marketing.txt ● PE.txt
New student → Add new student	This table saves the inputted information of a new student and stores them in a file. Creates file: ● New.txt
New student → Edit new student information	This table allows changes to be made on the saved information of a new student, identified by ID. Reads and writes file: ● New.txt
Visitor → Add new visitor	This table saves the inputted information of a visitor who wants to visit the school.

	Creates file: ● Visitor.txt
Visitor → Edit visitor information	This table allows the admin to change the information of a selected visitor request. Reads and writes file: ● Visitor.txt
Sports → add new sport information	This table allows a new sport event to be added to the database with the numbers of volunteers needed for each section. Creates file: ● Athlete.txt
Sports → Edit sport information	This table saves the changes made to the records of the existing sports events. Reads and writes file: ● Athlete.txt
Match → Match volunteers and requests	This table allows the admin to match a request with volunteers who are eligible to help (automatically filtered by system) Reads files: ● Volunteer.txt ● New.txt ● Visitor.txt ● Athlete.txt Writes file: ● Match.txt
Complete → Mark a matched event as complete	This table marks a match activity as complete. Reads file: ● Match.txt Writes file: ● Complete.txt
Report → generate report of the performance of a volunteer	This table allows the user to select the ID of a volunteer and view the report, including how many times the volunteer has helped and detailed information for each activity

	Reads file: ● Complete.txt

Relationship diagram:

I use the following relationship diagram to demonstrate the links between files in the relational database.

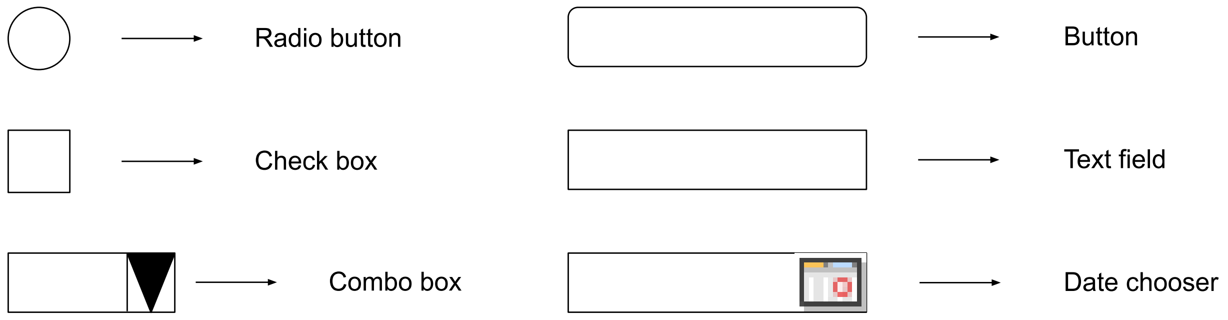


*Specific schemas of each table will be explained in the next section along with the prototype design and flow diagrams

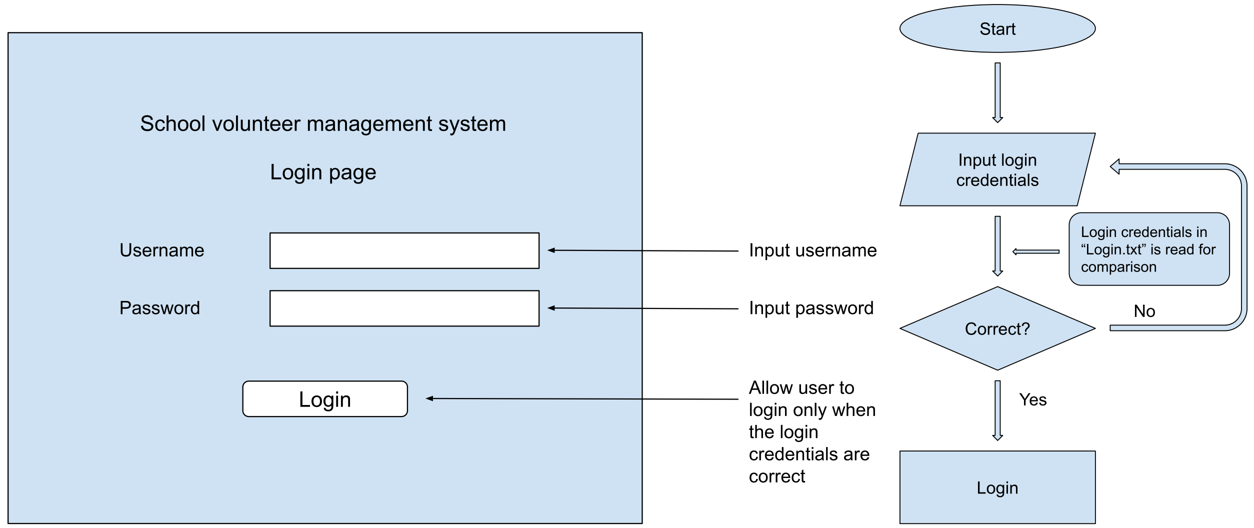
Prototype design with record structure and flow diagram:

The diagrams below are illustrations of the modules that will be created in my program. Each of them will serve a particular function and will be linked to the homepage.

Keys to be used in the modules:



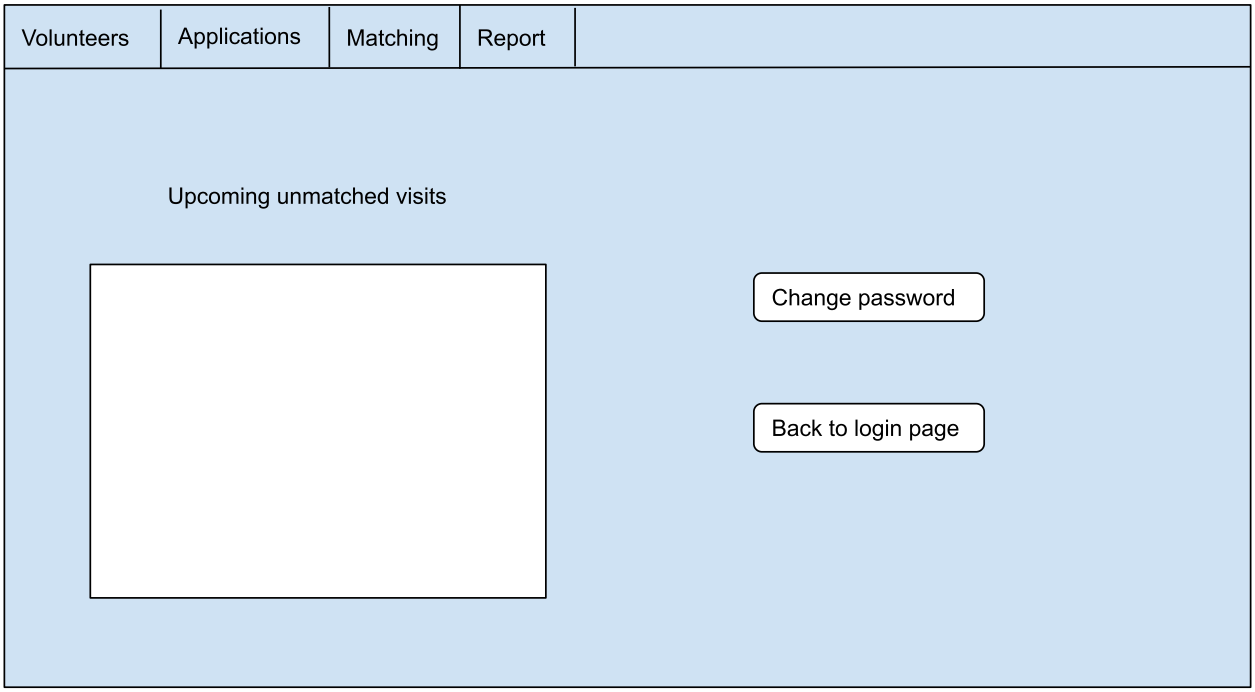
Login page


School volunteer management system
Login page
Username
Password
Login
Input username
Input password
Allow user to login only when the login credentials are correct
Start
Input login credentials
Login credentials in "Login.txt" is read for comparison
Correct?
No
Yes
Login

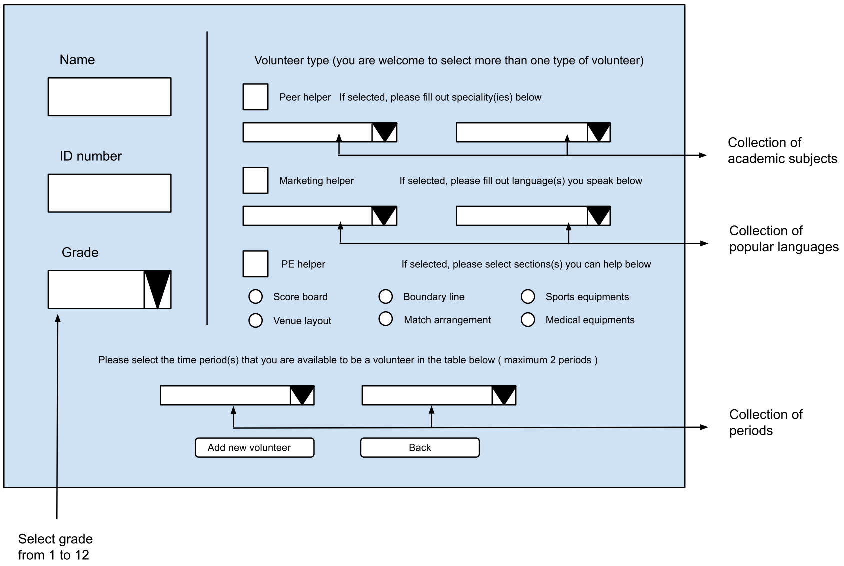
Login credential file

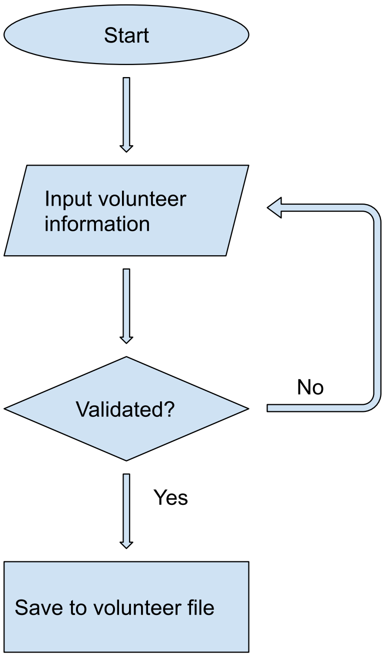
Field name	Data type	Description	Other information
Username	String	Username Example: “Brock”	/
Password	String	Password Example: “brock”	/

Home page



Add new volunteer page


Name
ID number
Grade
Volunteer type (you are welcome to select more than one type of volunteer)
□ Peer helper If selected, please fill out speciality(ies) below
□ Marketing helper If selected, please fill out language(s) you speak below
□ PE helper If selected, please select sections(s) you can help below
○ Score board
○ Boundary line
○ Sports equipments
○ Venue layout
○ Match arrangement
○ Medical equipments
Please select the time period(s) that you are available to be a volunteer in the table below (maximum 2 periods)
Add new volunteer
Back
Collection of academic subjects
Collection of popular languages
Collection of periods
Select grade from 1 to 12



**Editing volunteer page is the same except that the ID is a combobox for selection, the "Add new volunteer" becomes "Edit volunteer info", and a "Delete volunteer" button is added.*

Volunteer general data file

Field name	**Data type**	**Description**	**Other information**
Name	String	Volunteer’s name	/
ID	String	Volunteer’s ID	Primary key
Email	String	Volunteer’s email	Must contain “@”
Grade level	Integer	Grade level from 1 to 12	/
Available period 1	String	Date plus time slot Example: “Monday period 1”	/
Available period 2	String	Same as available period 1	Cannot be identical to period 1

Peer helpers data file

Field name	**Data type**	**Description**	**Other information**
ID	String	Volunteer’s ID	Secondary key copied from the ID in the volunteer general information file
Coaching subject 1	String	1st selected subject	/
Coaching subject 2	String	2nd selected subject	Cannot be identical to subject 1

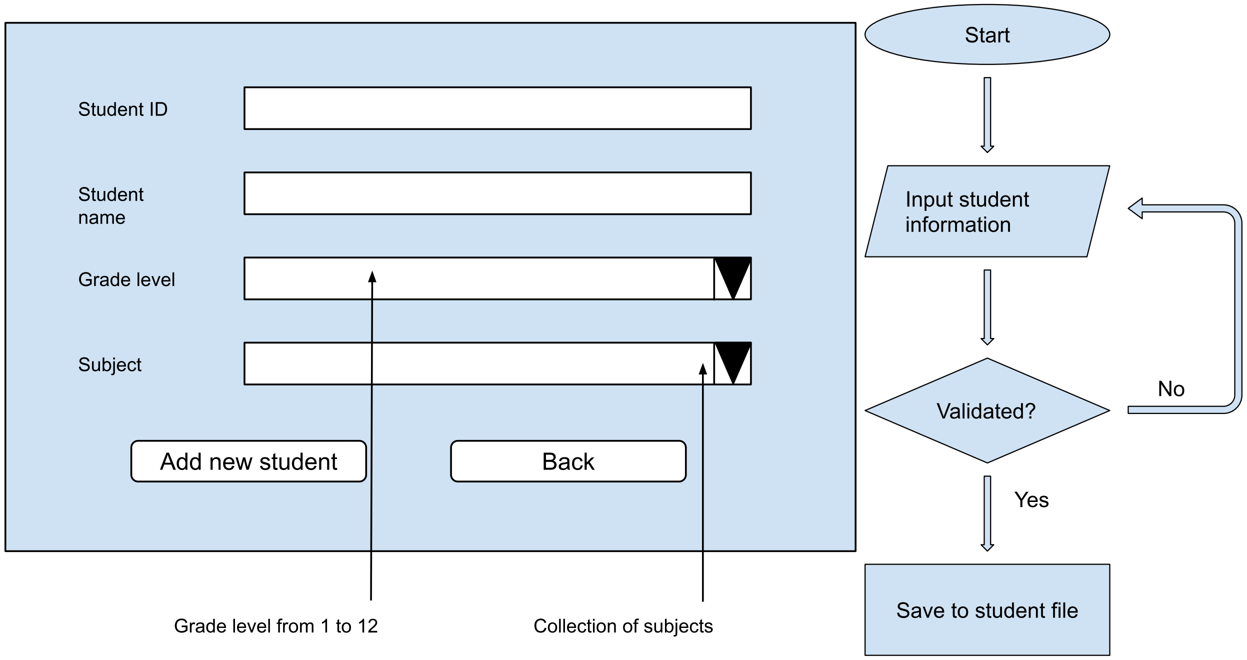
Marketing helpers data file

Field name	**Data type**	**Description**	**Other information**
ID	String	Volunteer’s ID	Secondary key copied from the ID in the volunteer general information file
Spoken language 1	String	1st selected language	/
Spoken language 2	String	2nd selected language	Cannot be identical to the 1st language

PE helpers data file

Field name	**Data type**	**Description**	**Other information**
ID	String	Volunteer's ID	Secondary key copied from the ID in the volunteer general information file
Score board	Integer	A number that indicates whether the volunteer helps with this section: "1" for yes and "0" for no.	Only one section among the 6 could be selected, which means there should be a "1" and five "0"s.
Boundary line	Integer	Same as above	
Sports equipments	Integer	Same as above	
Venue layout	Integer	Same as above	
Match arrangement	Integer	Same as above	
Medical equipments	Integer	Same as above	

Add new student page

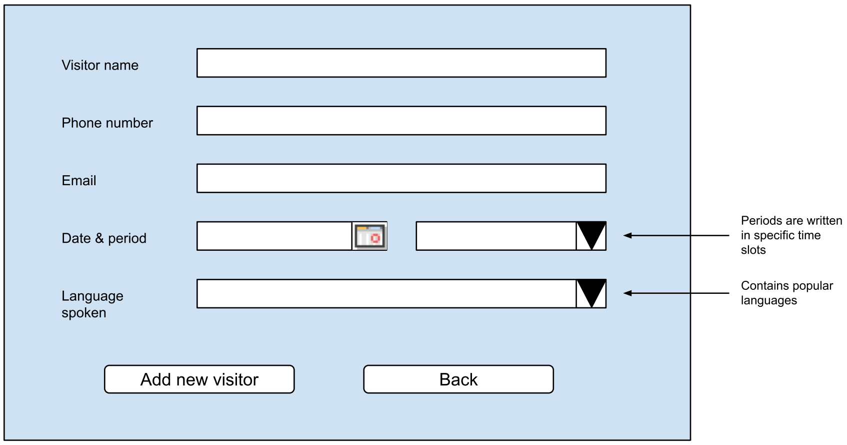


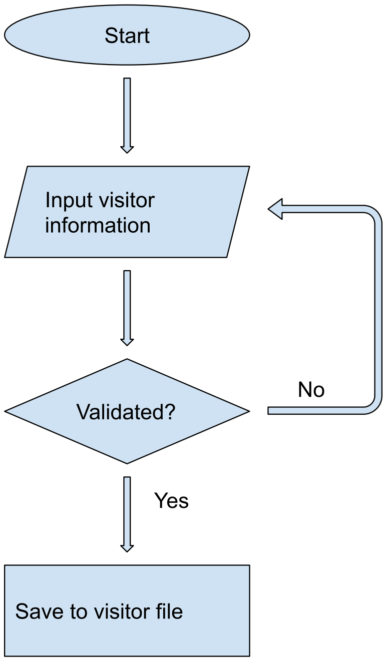
**Editing student page is the same except that the ID is a combobox for selection, the "Add new student" becomes "Edit student info", and a "Delete student" button is added.*

New student data file

Field name	**Data type**	**Description**	**Other information**
ID	String	New student's ID	Primary key
Name	String	New student's name	/
Email	String	New student's email	Must contain "@"
Grade level	Integer	Grade level from 1 to 12	/
Subject that need help with	String	Subject that is the student's weakness	/

Add visitor information page



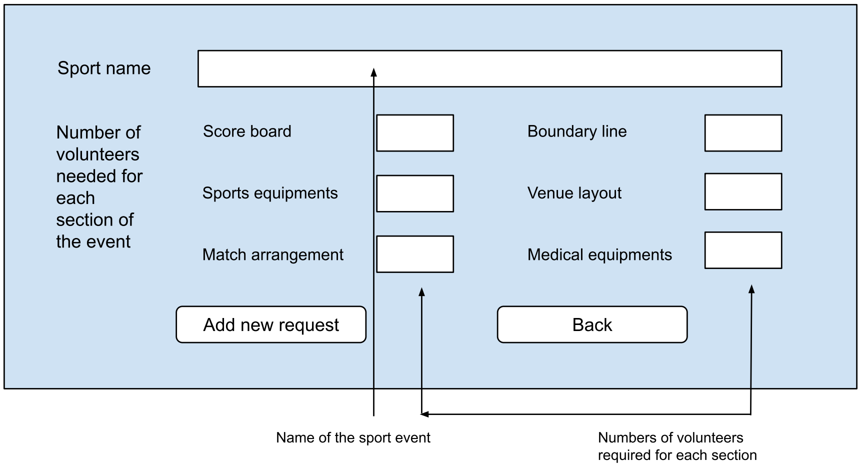


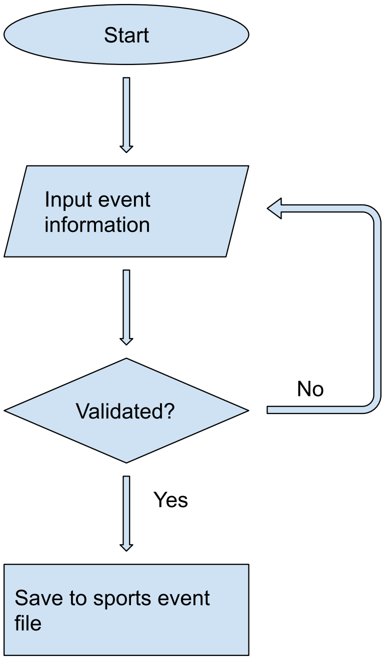
**Editing visitor page is the same except that the email is a combobox for selection, the "Add new visitor" becomes "Edit visitor info", and a "Delete visitor" button is added.*

Visitor data file

Field name	**Data type**	**Description**	**Other information**
Name	String	Name of the visitor	Primary key
Phone number	Integer	Phone number of the visitor	/
Email	String	Email address of the visitor	Primary key Must contain “@”
Date	String	Date when the visitor is going to come for a visit	/
Period	String	Time slot when the visitor is coming for a visit	/
Language	String	Language that the visitor prefer to speak	/

Add sport event page



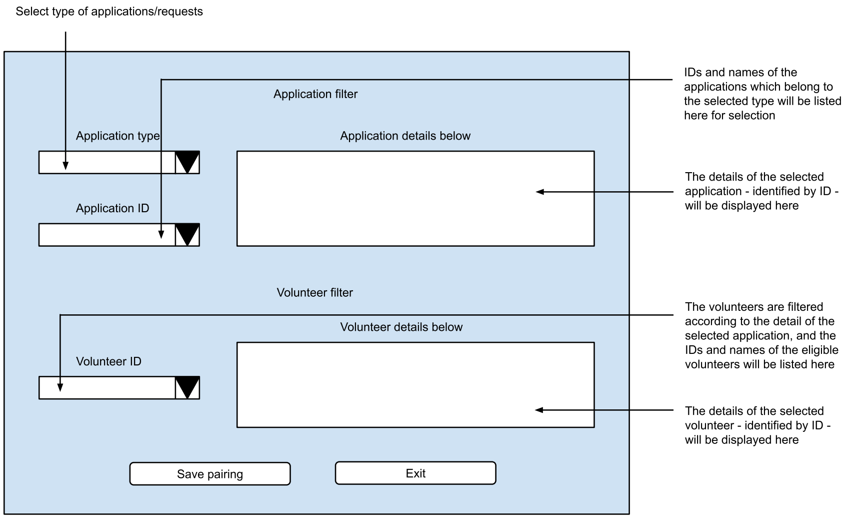


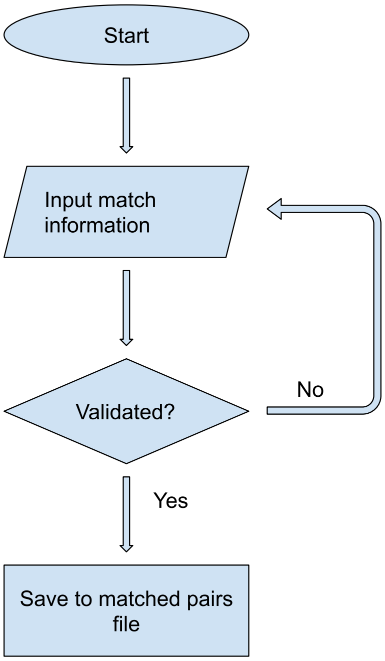
**Editing sport event page is the same except that the name is a combobox for selection, the “Add new request” becomes “Edit event info”, and a “Delete event” button is added.*

Sports events data file

Field name	Data type	Description	Other information
Sport name	String	Name of the sport	Primary key
Score board	Integer	Number of volunteers to help with flipping score boards	Larger than or equal to 0
Boundary line	Integer	Number of volunteers to help with setting up boundaries	Larger than or equal to 0
Sports equipments	Integer	Number of volunteers to help with managing sports equipments	Larger than or equal to 0
Venue layout	Integer	Number of volunteers to help with venue layout	Larger than or equal to 0
Match arrangement	Integer	Number of volunteers to help with arranging different competitions	Larger than or equal to 0
Medical equipments	Integer	Number of volunteers to help with managing medical equipments	Larger than or equal to 0

Matching page

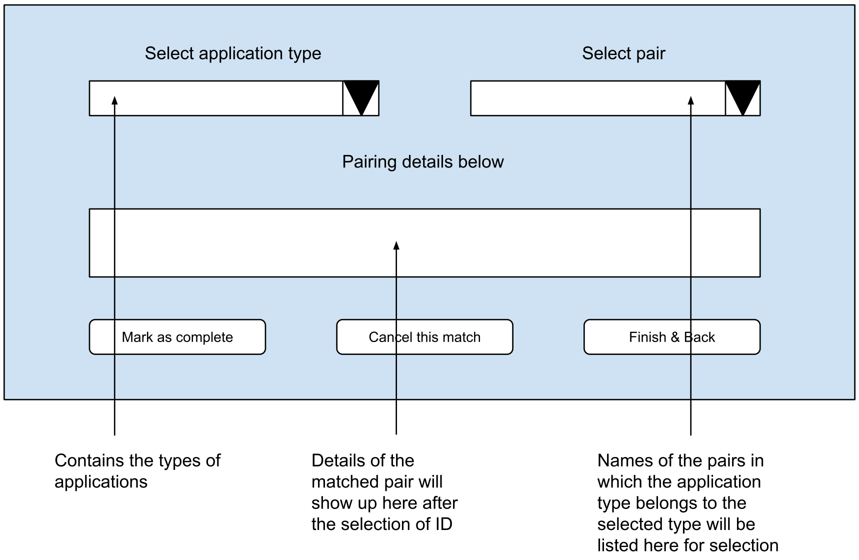

Select type of applications/requests
Application filter
Application type
Application details below
Application ID
Volunteer filter
Volunteer details below
Volunteer ID
Save pairing
Exit
IDs and names of the applications which belong to the selected type will be listed here for selection
The details of the selected application - identified by ID - will be displayed here
The volunteers are filtered according to the detail of the selected application, and the IDs and names of the eligible volunteers will be listed here
The details of the selected volunteer - identified by ID - will be displayed here



Successfully matched pairs file

Field name	**Data type**	**Description**	**Other information**
ID	String	ID for this match	Primary key
Type	String	Type of the application	/
Volunteer ID	String	ID of the volunteer	Secondary key copied from volunteer file
Student ID (for peer request) / Visitor email (for visitor request) / Sport name (for sport events)	String	Identifies the application/request	/

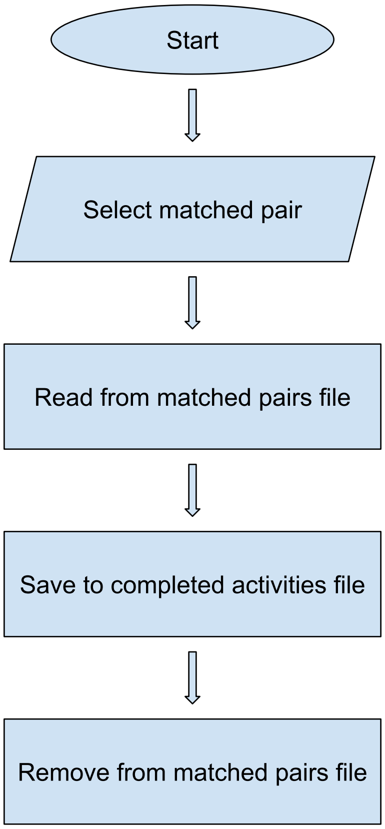
Mark completion page



Contains the types of applications

Details of the matched pair will show up here after the selection of ID

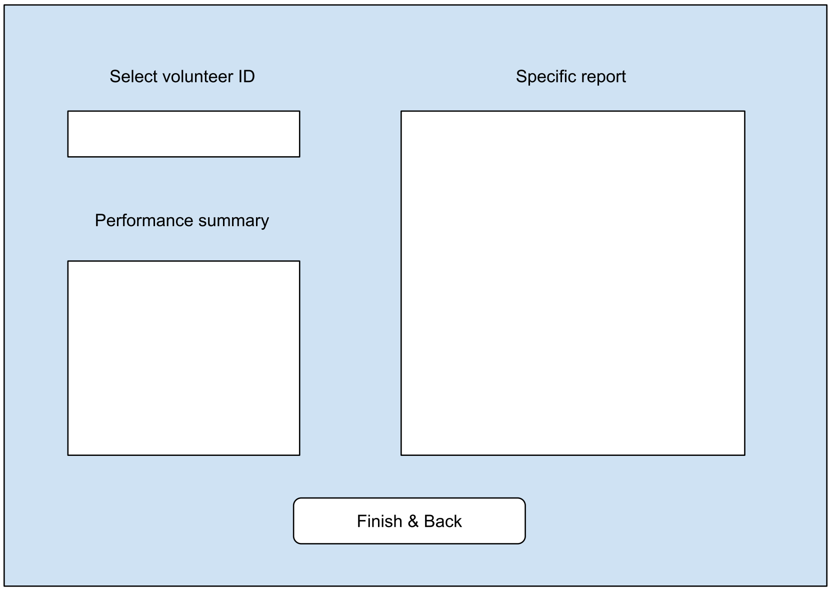
Names of the pairs in which the application type belongs to the selected type will be listed here for selection

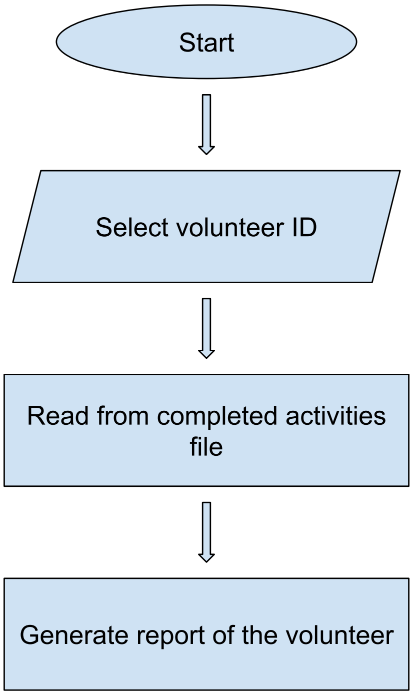

Start
Select matched pair
Read from matched pairs file
Save to completed activities file
Remove from matched pairs file

Completed activities file

Field name	**Data type**	**Description**	**Other information**
ID	String	ID for this match	Primary key
Type	String	Type of the application	/
Volunteer ID	String	ID of the volunteer	Secondary key copied from volunteer file
Student ID (for peer request) / Visitor email (for visitor request) / Sport name (for sport events)	String	Identifies the application/request	/

Generate volunteer report page




Start
Select volunteer ID
Read from completed activities file
Generate report of the volunteer

Tests required:

Module name	Nature of test	Example
Login → login information	Allows the user to enter the system if there is no error. Validation checks: ● The username and password tally with the correct credentials. If not then an error message should pop up and the user shouldn’t be allowed to login. Check method(s): ● Data verification	● The user input password as “123” while the correct password is “abc”
Volunteer → Add new volunteer	Allows the user to add a volunteer’s data if there is no error. Go to “Volunteer.txt”, “Peer.txt”, “Marketing.txt” and “PE.txt” to verify that data is successfully stored. Validation checks: ● Volunteer’s name, ID, grade level and at least 1 available period are selected ○ Volunteer’s name needs to be a string ○ Volunteer’s ID could not be the same as an existing volunteer ● If “Peer helper” is selected, at least one speciality should be selected ○ Specialities cannot be repeated ● If “Marketing helper” is selected, at least one	● The user leaves a field blank ● The user inputs “123” as a name ● The user selects “Peer helper” but selects no speciality

	language should be selected ○ Languages cannot be repeated ● If “PE helper” is selected, at least one section should be selected Check method(s): ● Completeness check ● Format check ● Consistency check ● Uniqueness check	
Volunteer → Edit volunteer information	Same as the add new volunteer module except that volunteer’s ID is going to be selected. Go to “Volunteer.txt”, “Peer.txt”, “Marketing.txt” and “PE.txt” to verify that data is successfully modified.	/
New student → Add new student	Allows the user to add a new student’s data if there is no error. Go to “New.txt” to verify that data is successfully stored. Validation checks: ● The student ID, name, grade level and subject that need help with are complete ○ Student’s name should be a string ○ Student’s ID could not be the same as an existing student Check method(s): ● Completeness check ● Format check ● Uniqueness check	● The user leaves a field blank ● The user inputs “123” as name

New student → Edit new student information	Same as the add new student module except that student’s ID is going to be selected. Go to “New.txt” to verify that data is successfully modified.	/
Visitor → Add new visitor	Allows the user to add a visitor’s data if there is no error. Go to “Visitor.txt” to verify that data is successfully stored. Validation checks: ● The visitor’s name, phone number, email, date, period and preferred language are complete ○ Visitor’s name should be a string ○ Phone number should contain digits only Check methods: ● Completeness check ● Format check	● The user leaves a field blank ● The user inputs “123” as name ● The user inputs “abc” as phone number
Visitor → Edit visitor information	Same as the add new visitor module except that visitor’s email is going to be selected. Go to “Visitor.txt” to verify that data is successfully modified.	/
Sports → add new sport information	Allows the user to add a sports event’s data if there is no error. Go to “Athlete.txt” to verify that data is successfully stored. Validation checks: ● The sport event’s name and numbers of volunteers required for each section are complete ○ The name of the event should be a	● The user leaves a field blank ● The user inputs a name that has already been used ● The user inputs “a” as a number of volunteers required

	string and should not be the same as any existing names ○ The numbers of volunteers required for each section should be integers larger than or equal to 0 Check method(s) - Completeness check - Format check - Range check - Uniqueness check	
Sports → Edit sport information	Same as the add new sport module except that sport's name is going to be selected. Go to "Athlete.txt" to verify that data is successfully modified.	/
Match → Match volunteers and requests	Allows the user to save a pair if there is no error. Go to "Match.txt" to verify that data is successfully stored. Validation checks: - A specific application is filtered and displayed on the table - An eligible volunteer is displayed on the table - A matching ID is inputted ○ The ID could not be the same as any existing matching IDs Check method(s) - Completeness check - Uniqueness check	- The user inputs an ID that is identical to an existing ID - The user forgot to select a volunteer for the application before saving

Complete → Mark a matched event as complete	Allows the user to mark a pair as complete. Go to “Completed.txt” to verify that data is successfully stored, and go to “Match.txt” to verify that the same record has been removed.	/
Report → generate report of the performance of a volunteer	Allows the user to view a volunteer’s performance after selecting a volunteer. No checking methods since no input needs to be checked	/