Criteria B: Design

Contents
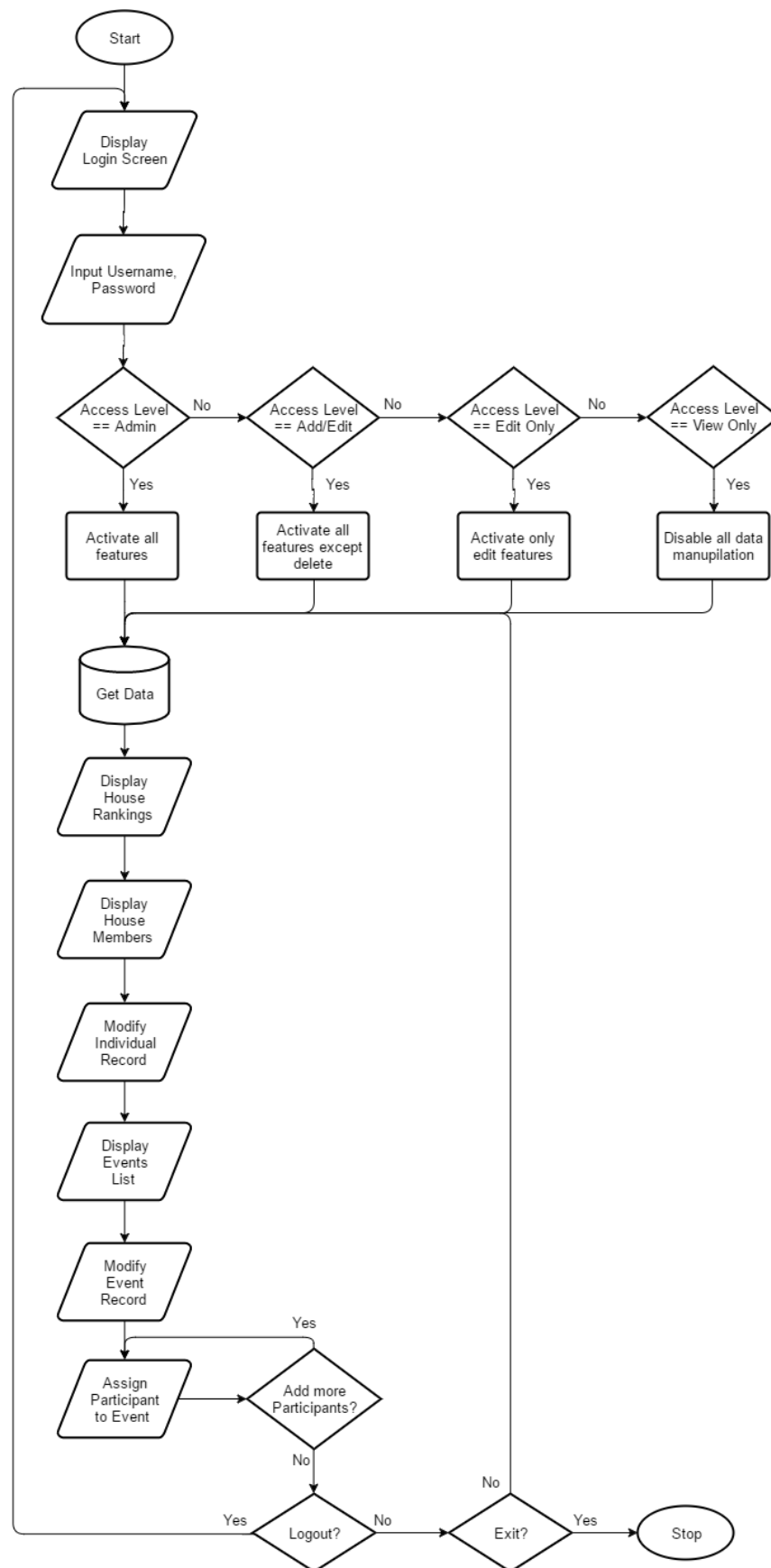
System Flow Chart

Figure 1: System Flow Chart for program
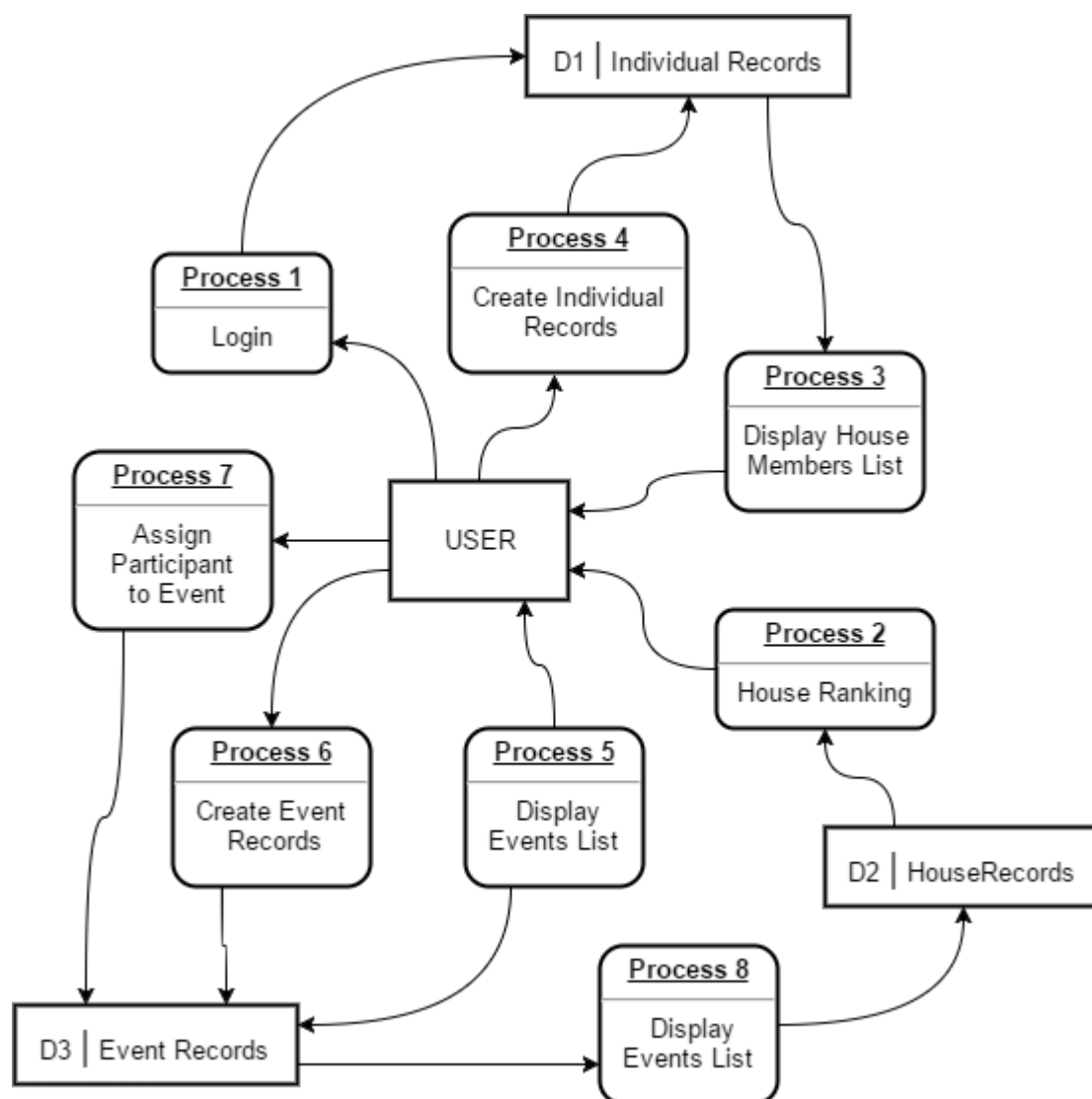
Data Flow Diagram

Figure 2: Data Flow Diagram for program

Planned Designs

The design of the application is done in a way to minimise the time taken to enter data. Data modification buttons are visible on their respective screens. The only buttons hidden in a menu are: logout, settings and exit; each of which should not be accedentally pressed.

Figure 3: Login Screen
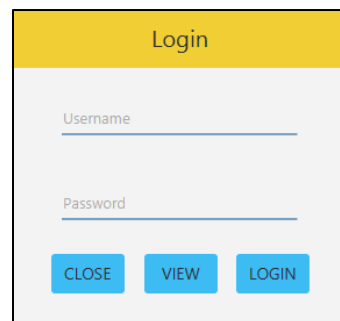

Figure 4: Home Screen – House Rankings
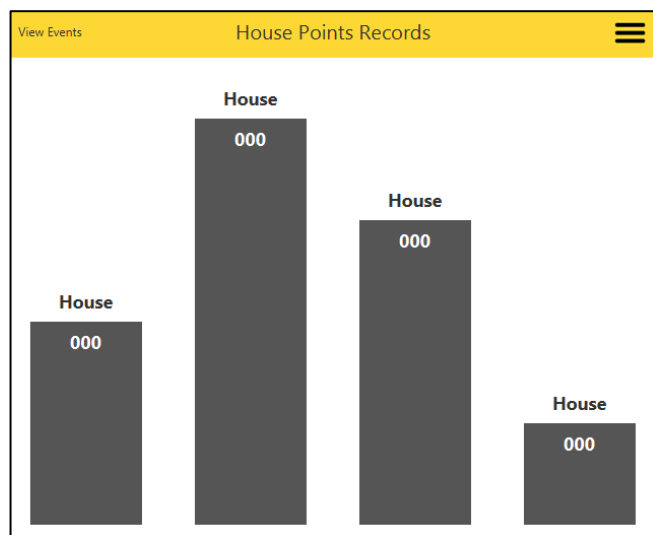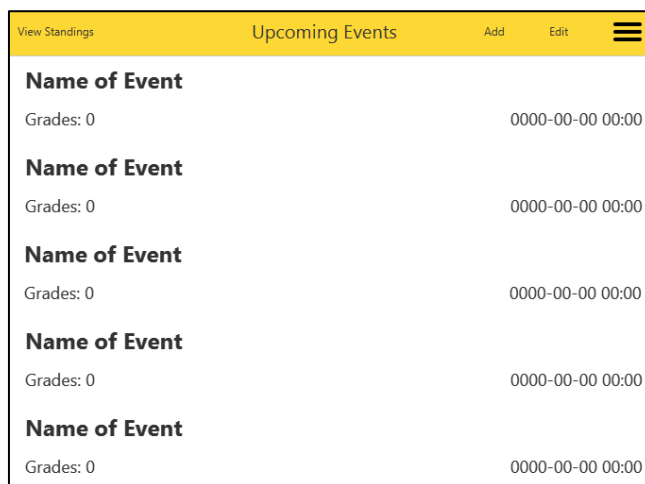
Figure 5: Upcoming Events List
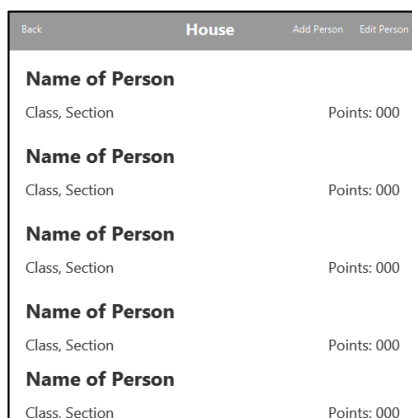
Figure 7: House Members
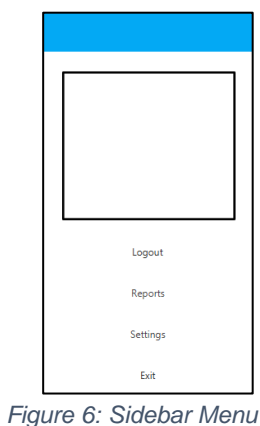
Figure 6: Sidebar Menu
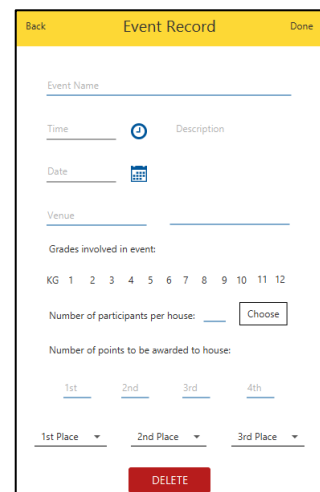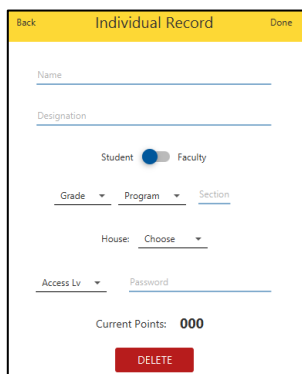
Figure 9: Event Record Editing

Figure 8: Individual Record Editing
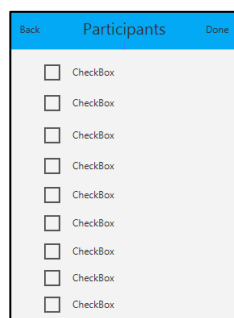
Figure 10: Participants Selection

Database Tables (backend storage)

Events		
Field Name	**Data Type**	**Role**
id	SMALLINT	Primary Key
name	VARCHAR(45)	Event Name
datetime	DATETIME	Event date and time
venue	VARCHAR(45)	Event Venue
grades	BIGINT	Grades involved in event
points1	TINYINT	1st position points
points2	TINYINT	2nd position points
points3	TINYINT	3rd position points
points4	TINYINT	4th position points
win1	SMALLINT	1st position house
win2	SMALLINT	2nd position house
win3	SMALLINT	3rd position house
win4	SMALLINT	4th position house
description	TEXT	Event Description

House		
Field Name	**Data Type**	**Properties**
id	SMALLINT	Primary Key
name	VARCHAR(20)	House Name
points	TINYINT	Total House Points
color	VARCHAR(7)	House Colour

Individual		
Field Name	**Data Type**	**Properties**
id	SMALLINT	Primary Key
name	VARCHAR(45)	Person Name
designation	VARCHAR(45)	Person Designation
house	VARCHAR(20)	Person House
is_student	BIT	If person is student (1) or faculty (0)
grade	TINYINT	If Student, grade
section	VARCHAR	If Student, section
program	BIT(2)	If faculty, program – PYP (1), MYP (2), DYP (3)
permission	BIT(2)	Access Level Permissions – View only (0), Edit only (1), Add/Edit (2), Admin (3)
pass_hash	CHAR(64)	SHA256 Hash of password
points	SMALLINT	Student Points

Participants		
Field Name	**Data Type**	**Properties**
id	INT	Primary Key
event	SMALLINT	Foreign Key (Event ID)
participant	SMALLINT	Foreign Key (Individual ID)

UML Diagrams

MainControlle

- AnchorPane base
- StackPane dialogStack
- Pane paneLogin
- JFXTextField fieldUser_Login
- JFXPasswordField fieldPass_Login
- JFXButton buttonLogin_Login
- JFXButton buttonClose_Login
- JFXButton buttonView_Login
- JFXHamburger buttonMenu
- JFXDrawer drawerMenu
- Pane paneEvents
- JFXButton buttonEdit_Events
- JFXButton buttonAdd_Events
- JFXListView<Pane> listView_Events
- JFXButton buttonRefresh
- Pane paneHome
- Pane podium1_Home
- Label name1_Home
- Label points1_Home
- Pane podium2_Home
- Label name2_Home
- Label points2_Home
- Pane podium3_Home
- Label name3_Home
- Label points3_Home
- Pane podium4_Home
- Label name4_Home
- Label points4_Home
- int[] listCellNo
- ~static Pane visiblePane
- ~static int accessLevel
- ~String color1
- ~String color2
- ~String color3
- ~String color4

+void initialize(URL url, ResourceBundle rb)
- void accessLevel(String level)
+void sqlDisconnect()
- void doSettings()
+void populateHouseRank()
- void populateEventList()
- void launchEventModify()
- String houseFromInt(int house)
- void launchEventModify(int id)
- void viewSwitch(ActionEvent event)
- void houseMembers(MouseEvent event)
- void popOutStandings(ActionEvent event)

PSGEventsManagementSyste

+void start(Stage stage)
+static void main(String[] args)

HamburgerDrawerControlle

~JFXButton buttonClose
~JFXButton buttonLogout
~JFXButton buttonSettings

+void initialize(URL url, ResourceBundle rb)

HouseMembersControll

- Pane base
- Label house
- JFXListView<Pane> listPerson
- JFXButton buttonEditPerson
- JFXButton buttonAddPerson
- JFXButton buttonRefresh
- int[] listCellNo

+void initialize(URL url, ResourceBundle rb)
+void setHouseName(String houseName)
+void setHouseColor(String houseColor)
- void launchPersonEditor()
- String programFromInt(int program)
- String accessFromInt(int accessLvl)
- void launchPersonEditor(int id)
- void populateList(String house)
- void buttonBack(ActionEvent event)

HouseStandingsControll

- AnchorPane base
- AnchorPane podium3
- Label name3
- Label points3
- AnchorPane podium1
- Label name1
- Label points1
- AnchorPane podium2
- Label name2
- Label points2
- AnchorPane podium4
- Label name4
- Label points4

+void initialize(URL url, ResourceBundle rb)
+void populateHouseRank()

SQL

~static Connection conn
~static boolean firstRun
~static MainController mc

+SQL()
+SQL(String dbName)
+static String login(String user, String pass)
+static void close()
+static String[][] houseRank()
+static ResultSet eventList()
+static ResultSet personList(String house)
+static ResultSet participantList()
+static ResultSet participantSelection(int id)
+static void addParticipants(int event, int participant)
- static int houseInt(String house)
+static void addEvent(String name, String datetime, String venue, int grades, int points1, int points2, int points3, int points4, String win1, String win2, String win3, String win4, String description)
+static void updateEvent(int id, String name, String datetime, String venue, int grades, int points1, int points2, int points3, int points4, String win1, String win2, String win3, String win4, String description)
+static void deleteEvent(int id)
+static ResultSet loadEvent(int id)
- static void updatePoints(int EvID, int win1, int point1, int win2, int point2, int win3, int point3, int win4, int point4)
+static ResultSet loadParticipants(int id)
+static void addPerson(String name, String designation, String house, boolean isFaculty, int grade, char section, int program, int permissions, String pass)
+static void updatePerson(int id, String name, String designation, String house, boolean isFaculty, int grade, char section, int program, int permissions, String pass)
+static void deletePerson(int id)
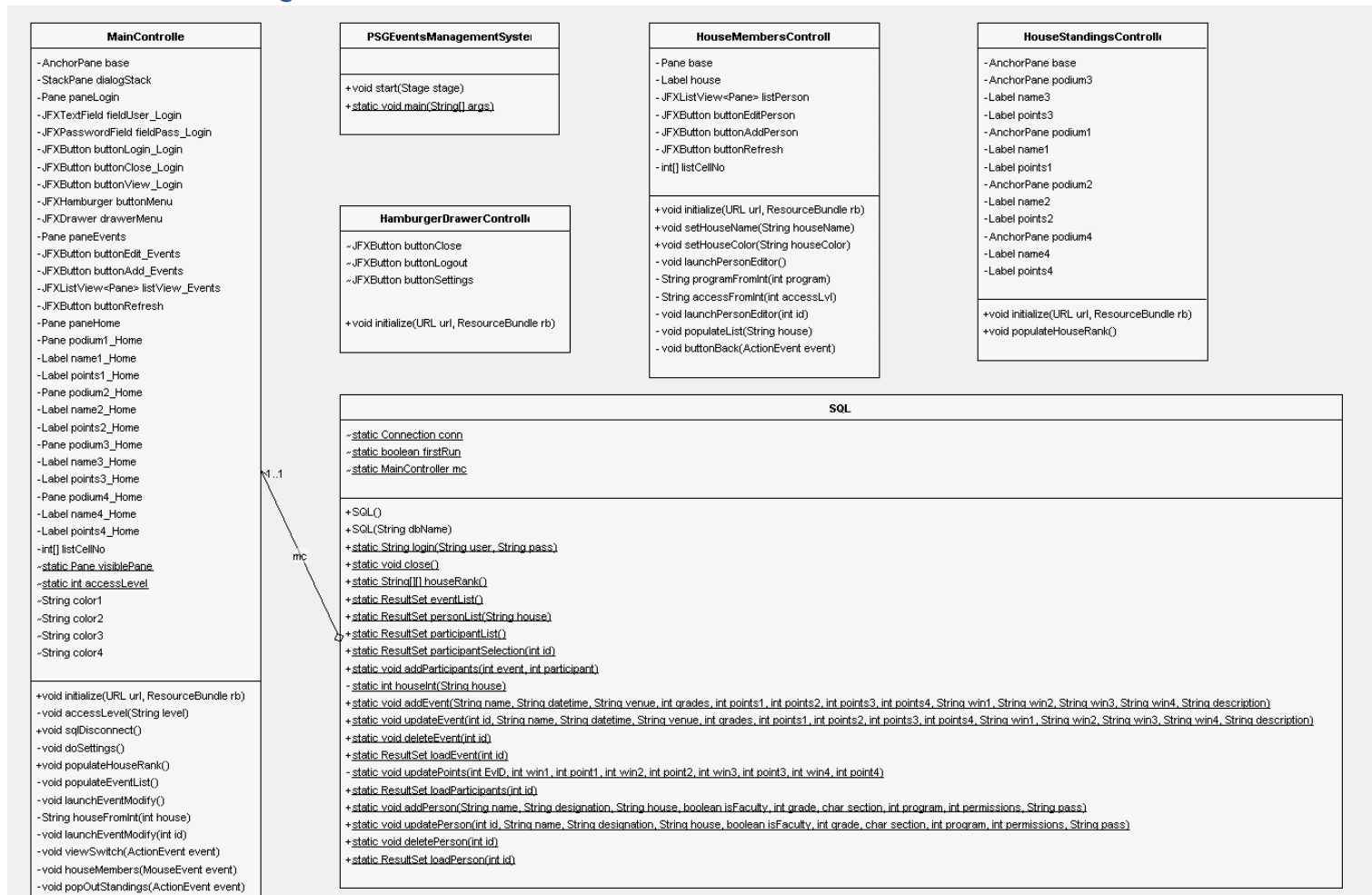+static ResultSet loadPerson(int id)

1..1 mc

Figure 11: UML Diagram – PSGEventManagementSystem, MainController, HamburgerDrawerController, HouseMembersController, HouseStandingsController, SQL.
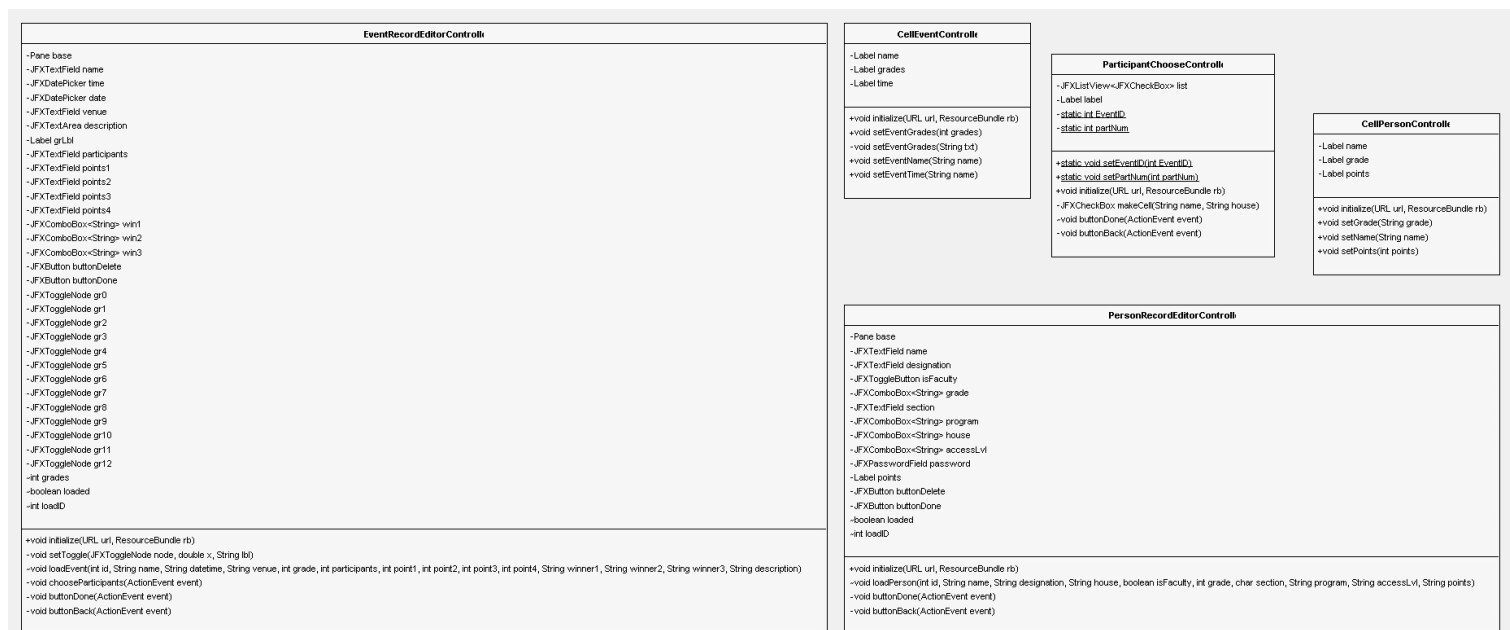
EventRecordEditorControlle

- Pane base
- JFXTextField name
- JFXDatePicker time
- JFXDatePicker date
- JFXTextField venue
- JFXTextArea description
- Label grLbl
- JFXTextField participants
- JFXTextField points1
- JFXTextField points2
- JFXTextField points3
- JFXTextField points4
- JFXComboBox<String> win1
- JFXComboBox<String> win2
- JFXComboBox<String> win3
- JFXButton buttonDelete
- JFXButton buttonDone
- JFXToggleNode gr0
- JFXToggleNode gr1
- JFXToggleNode gr2
- JFXToggleNode gr3
- JFXToggleNode gr4
- JFXToggleNode gr5
- JFXToggleNode gr6
- JFXToggleNode gr7
- JFXToggleNode gr8
- JFXToggleNode gr9
- JFXToggleNode gr10
- JFXToggleNode gr11
- JFXToggleNode gr12
- int grades
- boolean loaded
- int loadID

+void initialize(URL url, ResourceBundle rb)
- void setToggle(JFXToggleNode node, double x, String lbl)
- void loadEvent(int id, String name, String datetime, String venue, int grade, int participants, int point1, int point2, int point3, int point4, String winner1, String winner2, String winner3, String description)
- void chooseParticipants(ActionEvent event)
- void buttonDone(ActionEvent event)
- void buttonBack(ActionEvent event)

CellEventControll

- Label name
- Label grades
- Label time

+void initialize(URL url, ResourceBundle rb)
+void setEventGrades(int grades)
- void setEventGrades(String txt)
+void setEventName(String name)
+void setEventTime(String name)

ParticipantChooseControll

- JFXListView<JFXCheckBox> list
- Label label
- static int EventID
- static int partNum

+static void setEventID(int EventID)
+static void setPartNum(int partNum)
+void initialize(URL url, ResourceBundle rb)
- JFXCheckBox makeCell(String name, String house)
- void buttonDone(ActionEvent event)
- void buttonBack(ActionEvent event)

CellPersonControlle

- Label name
- Label grade
- Label points

+void initialize(URL url, ResourceBundle rb)
+void setGrade(String grade)
+void setName(String name)
+void setPoints(int points)

PersonRecordEditorControlle

- Pane base
- JFXTextField name
- JFXTextField designation
- JFXToggleButton isFaculty
- JFXComboBox<String> grade
- JFXTextField section
- JFXComboBox<String> program
- JFXComboBox<String> house
- JFXComboBox<String> accessLvl
- JFXPasswordField password
- Label points
- JFXButton buttonDelete
- JFXButton buttonDone
- boolean loaded
- int loadID

+void initialize(URL url, ResourceBundle rb)
- void loadPerson(int id, String name, String designation, String house, boolean isFaculty, int grade, char section, String program, String accessLvl, String points)
- void buttonDone(ActionEvent event)
- void buttonBack(ActionEvent event)

Figure 12: EventRecordEditorController, CellEventController, ParticipantChooserController, PersonRecordEditorController, CellPersonController.

Test Plan

Criteria	Aim of Test	Example input	Expected Output
6	Account based access via login screen. Verifying administrator access	Admin Login: Username- admin Password- adminpassword	All buttons and fields editable
6	Account based access via login screen. Verifying view only user access.	User without editing access.	All data modifying buttons and fields disabled.
1	Display current house rankings	—	Name, points and colour of houses displayed in fig. 4.
2	Display upcoming events	—	Name, involvement and time displayed in fig. 5.
2	Display members of each house	—	Name, grade and points displayed in fig. 7.
4	Adding individual record containing name, designation, house, access level and password, and teacher's program.	Name: Srividya Designation: Events Coordinator House: Earth Access level: Add/Edit Password: (>= 10 char) Program: DYP	Can login using details entered with add/edit access
4	Adding individual record containing name, designation, house, access level and password, and student's grade and section.	Name: Prithvi Designation: Sports Captain House: Earth Access level: View Only Grade: 12 Section C	Added to Earth house members list.
3	Adding event record containing name, venue, date, time, grades involved, participants, points to be awarded, winners.	Name: 4x100m Relay U-19 Venue: Field Date: 4 Feb 2017 Time: 11:00 Grades: 9 to 12 Participants: 4 from each house Points: 10, 7, 3, 0	Event is added to list. Participants are assigned to the event.

1, 3	Editing event to add winners.	Event name: 4x100m Relay U-19 Winners: Earth, Fire, Air, Water	House points get added to total. Each participant gets points as per their house.
5	Generating report for testing period.	Month during which athletics events occur	Get report with events (winners, points), houses (points gained), top achieving people.