IT PAT

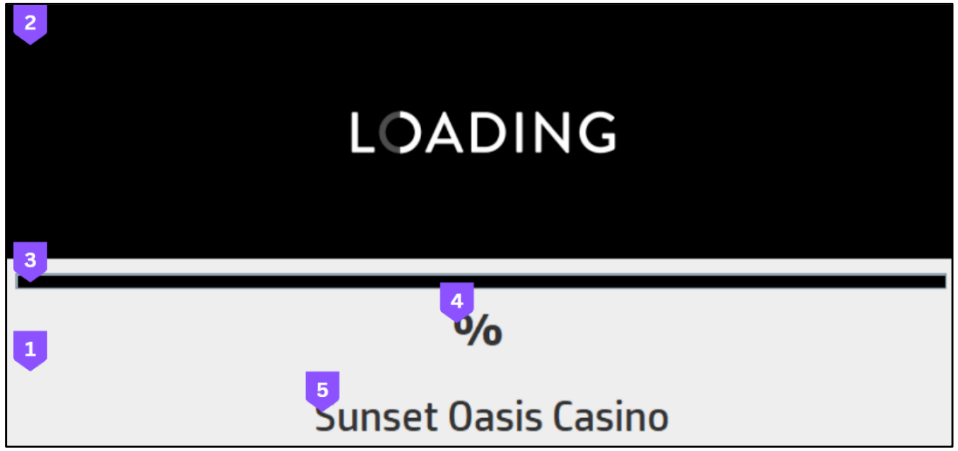
Phase 2: Design Documentation

Name: Milaan Kassie

Table of Contents

2.1 User Interface Design

Splash Screen

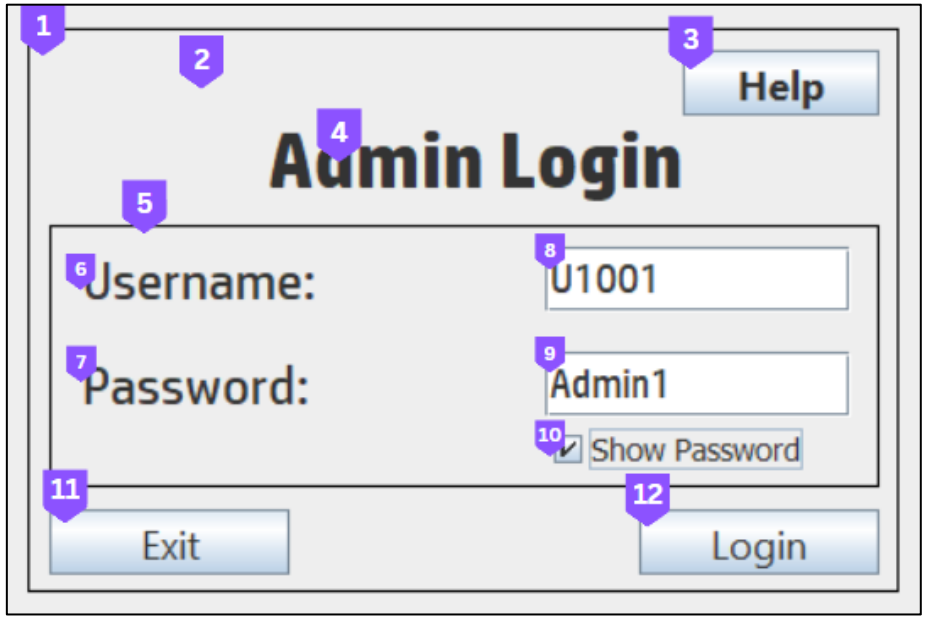


Splash screen components

- Layered Pane – Background
- Label - Loading GIF
- Label - Percent loaded
- Label - Company name
- Progress bar - Loading progress

NO	COMPONENT	INPUT	EVENT
0	Frame		
1	Layered pane		
2	Label		Plays the loading GIF
3	Progress bar		Loads until completion
4	Label		Increases percentage from 0 to 100
5	Label		

Login Frame

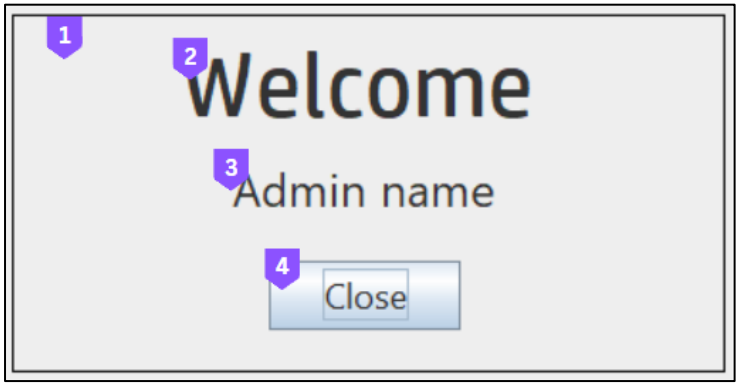


Login Frame components

- **Level 1: Layered Pane – Background**
  - Label - Title of screen (Admin Login)
  - Button - Help button in top right
  - Button - Exit button for closing application (leads to LogoutFrame)
  - Button - Login button (leads WelcomeFrame and then to MainMenuFrame)
    - **Level 2: Layered Pane - pane for the username and password**
      - Label - Username label
      - Text Field - Entry for username
      - Label - Password label
      - Text Field - Entry for password
      - Checkbox - show password checkbox

NO	COMPONENT	INPUT	EVENT
0	Frame		
1	Layered pane		
2	Label		
3	Button	Mouse	Brings up the Help Frame
4	Label		
5	Layered pane		
6	Label		
7	Label		
8	Text field	Keyboard	
9	Text field	Keyboard	
10	Check box	Mouse	Shows/Hides password
11	Button	Mouse	Leads up the Exit Frame to confirm closing the application
12	Button	Mouse	Logs user in if credentials are valid - Leads to Welcome Frame

Welcome Frame

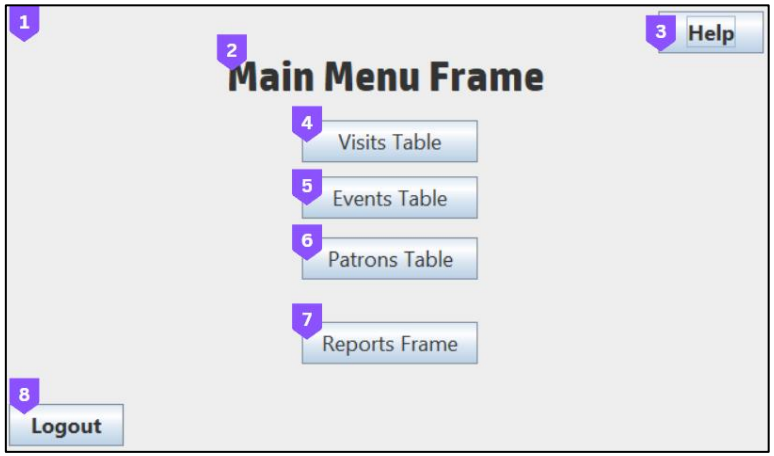


Welcome Frame Components

- Layered Pane – Background
- Label - Screen heading (welcome message)
- Label - Username (changes depending on user)
- Button - Close screen that closes this frame and leads to the MainMenuFrame

NO	COMPONENT	INPUT	EVENT
0	Frame		
1	Layered pane		
2	Label		
3	Label		
4	Button	Mouse	Closes the Welcome Frame and leads to the Main Menu

Main Menu

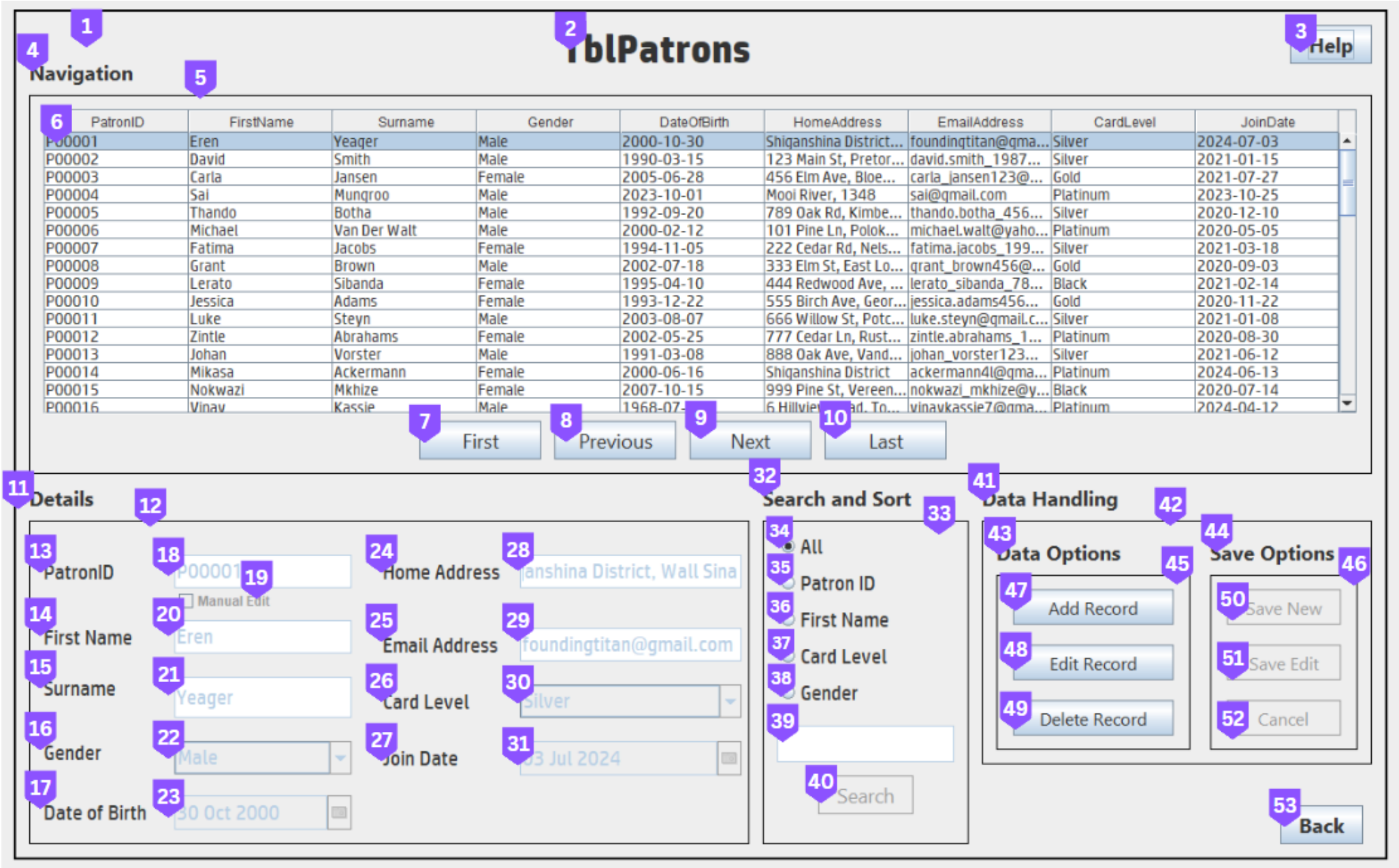


Main Menu Components

- Layered Pane – Background
  - Label - Screen heading/title (Main Menu Frame)
  - Button - Help button on top right
  - Button - VisitsTable
  - Button - EventsTable
  - Button - PatronsTable
  - Button - ReportsFrame
  - Button - Logout button

NO	COMPONENT	INPUT	EVENT
0	Frame		
1	Layered pane		
2	Label		
3	Button	Mouse	Brings up the Help Frame
4	Button	Mouse	Leads to the Visits Table Frame
5	Button	Mouse	Leads to the Events Table Frame
6	Button	Mouse	Leads to the Patrons Table Frame
7	Button	Mouse	Leads to the Reports Frame
8	Button	Mouse	Leads to the Admin Login Frame

Patrons Table



PatronID	FirstName	Surname	Gender	DateOfBirth	HomeAddress	EmailAddress	CardLevel	JoinDate
P00001	Eren	Yeager	Male	2000-10-30	Shiganshina District...	foundingtitan@gma...	Silver	2024-07-03
P00002	David	Smith	Male	1990-03-15	123 Main St, Pretor...	david.smith_1987...	Silver	2021-01-15
P00003	Carla	Jansen	Female	2005-06-28	456 Elm Ave, Bloe...	carla_jansen123@...	Gold	2021-07-27
P00004	Sai	Mungroo	Male	2023-10-01	Mooi River, 1348	sai@gmail.com	Platinum	2023-10-25
P00005	Thando	Botha	Male	1992-09-20	789 Oak Rd, Kimbe...	thando.botha_456...	Silver	2020-12-10
P00006	Michael	Van Der Walt	Male	2000-02-12	101 Pine Ln, Polok...	michael.walt@yaho...	Platinum	2020-05-05
P00007	Fatima	Jacobs	Female	1994-11-05	222 Cedar Rd, Nels...	fatima.jacobs_199...	Silver	2021-03-18
P00008	Grant	Brown	Male	2002-07-18	333 Elm St, East Lo...	grant_brown456@...	Gold	2020-09-03
P00009	Lerato	Sibanda	Female	1995-04-10	444 Redwood Ave, ...	lerato_sibanda_78...	Black	2021-02-14
P00010	Jessica	Adams	Female	1993-12-22	555 Birch Ave, Geor...	jessica.adams456...	Gold	2020-11-22
P00011	Luke	Steyn	Male	2003-08-07	666 Willow St, Potc...	luke.steyn@gmail.c...	Silver	2021-01-08
P00012	Zintle	Abrahams	Female	2002-05-25	777 Cedar Ln, Rust...	zintle.abrahams_1...	Platinum	2020-08-30
P00013	Johan	Vorster	Male	1991-03-08	888 Oak Ave, Vand...	johan_vorster123...	Silver	2021-06-12
P00014	Mikasa	Ackermann	Female	2000-06-16	Shiganshina District	ackermann4l@gma...	Platinum	2024-06-13
P00015	Nokwazi	Mkhize	Female	2007-10-15	999 Pine St, Vereen...	nokwazi_mkhize@y...	Black	2020-07-14
P00016	Vinay	Kassie	Male	1968-07-	6 Hillvie ad. To...	vinaykassie7@gma...	Platinum	2024-04-12

Patrons Table Components

- **Level 1: Layered Pane – Background**
  - Label - Heading/Title label (Visits Table)
  - Button - Help button in top right
  - Button - Back button (leads back to LoginFrame)

- **Level 2: Layered Pane - Navigation (contains JTable)**
  - JTable - PatronsTable
  - Button - First record
  - Button - Previous record
  - Button - Next record
  - Button - Last record

- **Level 2: Layered Pane - Details**
  - Label - PatronID
  - Text Field - PatronID
  - Label - PatronName
  - Text Field - PatronName
  - Label - Surname
  - Text Field - Surname
  - Label - Gender
  - Combo Box - Gender
  - Label - DateOfBirth
  - JDateChooser- DateOfBirth
  - Label - HomeAddress
  - Text Field - HomeAdress
  - Label - Status
  - Combo Box - Status
  - Label - Registration Deadline
  - DateChooser - Registration Deadline

- **Level 2: Layered Pane – Data Handling**
  - **Level 3: Layered Pane - Options**
    - Button - Add (allows data to be entered into the details pane for the add function)
    - Button - Edit (allows data to be edited in the details pane for the edit function)
    - Button - Delete (deletes the currently selected record)

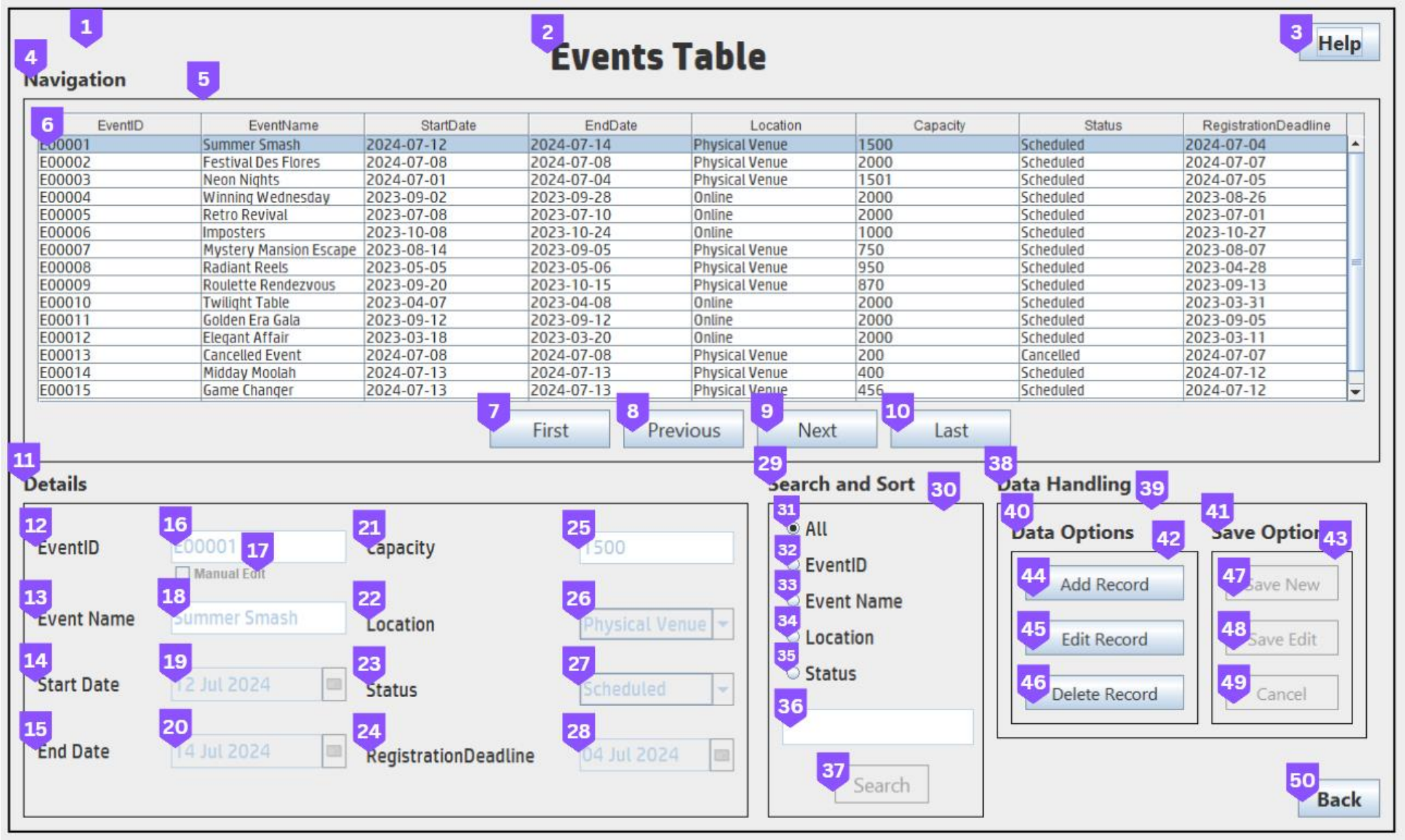
- **Level 3: Layered Pane - Save Options**
    - Button - Save new (saves data entered in the details pane by calling the SQL method)
    - Button - Save edit (saves the data entered in the details pane by calling the SQL method)
    - Button - Cancel (makes the details pane inaccessible and removes all data entered/edited)

- **Level 2: Layered Pane - Search**
    - Radio Button - All (shows all records)
    - Radio Button - PatronID (deactivates all other radio buttons and enables the text field for entry of a PatronID)
    - Radio Button - FirstNam (deactivates all other radio buttons and enables the text field for entry of a FirstName)
    - Radio Button - CardLevel (deactivates all other radio buttons and enables the text field for entry of a CardLevel)
    - Radio Button – Gender (deactivates all other radio buttons and enables the text field for entry of a CardLevel)

NO	COMPONENT	INPUT	EVENT
0	Frame		
1	Layered pane		
2	Label		
3	Button	Mouse	Brings up the Help Frame
4	Label		
5	Layered pane		
6	Table		
7	Button	Mouse	Navigates to and displays the first record in the table
8	Button	Mouse	Navigates to and displays the next record in the table
9	Button	Mouse	Navigates to and displays the previous record in the table
10	Button	Mouse	Navigates to and displays the last record in the table
11	Label		
12	Layered pane		
13	Label		
14	Label		
15	Label		
16	Label		
17	Label		
18	Text field	Keyboard	
19	Check box	Mouse	Enables / Disables primary key field for manual editing
20	Text field	Keyboard	
21	Text field	Keyboard	

NO	COMPONENT	INPUT	EVENT
22	Combo box	Mouse	
23	Date chooser	Mouse / Keyboard	
24	Label		
25	Label		
26	Label		
27	Label		
28	Text field	Keyboard	
29	Text field	Keyboard	
30	Combo box	Mouse	
31	Date chooser	Mouse / Keyboard	
32	Label		
33	Layered pane		
34	Radio button	Mouse	Disables search text field and button & displays all records in table
35	Radio button	Mouse	Enables search text field and button
36	Radio button	Mouse	Enables search text field and button
37	Radio button	Mouse	Enables search text field and button
38	Radio button	Mouse	Enables search text field and button
39	Text field	Keyboard	
40	Button	Mouse	Searches/sorts the table records according to the radio button selected and data entered
41	Label		
42	Layered pane		
43	Label		
44	Label		
45	Layered pane		
46	Layered pane		
47	Button	Mouse	Activates and clears relevant data entry fields
48	Button	Mouse	Activates and inputs current data into relevant data entry fields
49	Button	Mouse	Clears data and deactivates data entry fields - displays current data
50	Button	Mouse	Executes SQL to add a new record to the table using entered data
51	Button	Mouse	Executes SQL to edit a record in the table using the edited data
52	Button	Mouse	Executes SQL to delete a record in the table
53	Button	Mouse	Leads to the Main Menu Frame

Events Table



EventID	EventName	StartDate	EndDate	Location	Capacity	Status	RegistrationDeadline
E00001	Summer Smash	2024-07-12	2024-07-14	Physical Venue	1500	Scheduled	2024-07-04
E00002	Festival Des Flores	2024-07-08	2024-07-08	Physical Venue	2000	Scheduled	2024-07-07
E00003	Neon Nights	2024-07-01	2024-07-04	Physical Venue	1501	Scheduled	2024-07-05
E00004	Winning Wednesday	2023-09-02	2023-09-28	Online	2000	Scheduled	2023-08-26
E00005	Retro Revival	2023-07-08	2023-07-10	Online	2000	Scheduled	2023-07-01
E00006	Imposters	2023-10-08	2023-10-24	Online	1000	Scheduled	2023-10-27
E00007	Mystery Mansion Escape	2023-08-14	2023-09-05	Physical Venue	750	Scheduled	2023-08-07
E00008	Radiant Reels	2023-05-05	2023-05-06	Physical Venue	950	Scheduled	2023-04-28
E00009	Roulette Rendezvous	2023-09-20	2023-10-15	Physical Venue	870	Scheduled	2023-09-13
E00010	Twilight Table	2023-04-07	2023-04-08	Online	2000	Scheduled	2023-03-31
E00011	Golden Era Gala	2023-09-12	2023-09-12	Online	2000	Scheduled	2023-09-05
E00012	Elegant Affair	2023-03-18	2023-03-20	Online	2000	Scheduled	2023-03-11
E00013	Cancelled Event	2024-07-08	2024-07-08	Physical Venue	200	Cancelled	2024-07-07
E00014	Midday Moolah	2024-07-13	2024-07-13	Physical Venue	400	Scheduled	2024-07-12
E00015	Game Changer	2024-07-13	2024-07-13	Physical Venue	456	Scheduled	2024-07-12

Events Table Components

- **Level 1: Layered Pane - Cover whole screen to provide a border**
    - Label - Heading/Title label (Visits Table)
    - Button - Help button in top right
    - Button - Back button (leads back to LoginFrame)

- **Level 2: Layered Pane - Navigation (contains JTable)**
    - JTable - EventsTable
    - Button - First record
    - Button - Previous record
    - Button - Next record
    - Button - Last record

- **Level 2: Layered Pane - Details**
    - Label - EventID
    - Text Field - EventID
    - Label - EventName
    - Text Field - EventName
    - Label - StartDate
    - Date Chooser - StartDate
    - Label - EndDate
    - Date Chooser - EndDate
    - Label - Capacity
    - Text Field - Capacity
    - Label - Location
    - Combo Box - Location
    - Label - Status
    - Combo Box - Status
    - Label - Registration Deadline
    - DateChooser - Registration Deadline

- **Level 2: Layered Pane – Data Handling**
    - **Level 3: Layered Pane - Options**
        - Button - Add (allows data to be entered into the details pane for the add function)
        - Button - Edit (allows data to be edited in the details pane for the edit function)
        - Button - Delete (deletes the currently selected record)

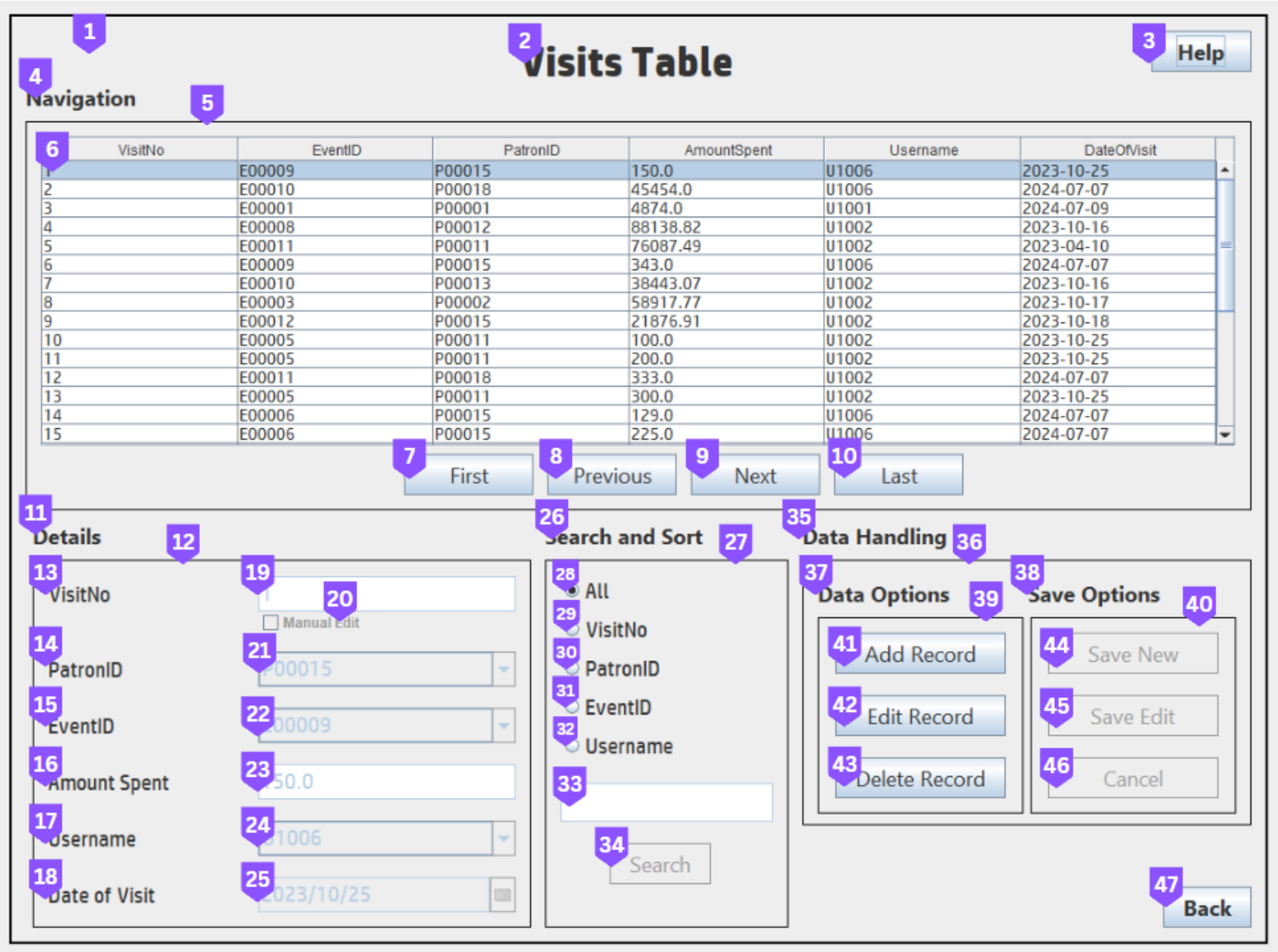
- **Level 3: Layered Pane - Save Options**
    - Button - Save new (saves data entered in the details pane by calling the SQL method)
    - Button - Save edit (saves the data entered in the details pane by calling the SQL method)
    - Button - Cancel (makes the details pane inaccessible and removes all data entered/edited)

- **Level 2: Layered Pane - Search**
    - Radio Button - All (shows all records)
    - Radio Button - EventID (deactivates all other radio buttons and enables the text field for entry of a EventNo)
    - Radio Button - EventName (deactivates all other radio buttons and enables the text field for entry of a EventName)
    - Radio Button - Location (deactivates all other radio buttons and enables the text field for entry of a Location)
    - Radio Button – Status (deactivates all other radio buttons and enables the text field for entry of a Status)

NO	COMPONENT	INPUT	EVENT
0	Frame		
1	Layered pane		
2	Label		
3	Button	Mouse	Brings up the Help Frame
4	Label		
5	Layered pane		
6	Table		
7	Button	Mouse	Navigates to and displays the first record in the table
8	Button	Mouse	Navigates to and displays the next record in the table
9	Button	Mouse	Navigates to and displays the previous record in the table
10	Button	Mouse	Navigates to and displays the last record in the table
11	Label		
12	Layered pane		
13	Label		
14	Label		
15	Label		
16	Text field	Keyboard	
17	Check box	Mouse	Enables / Disables primary key field for manual editing
18	Text field	Keyboard	
19	Date chooser	Mouse / Keyboard	
20	Date chooser	Mouse / Keyboard	
21	Label		

NO	COMPONENT	INPUT	EVENT
22	Label		
23	Label		
24	Label		
25	Text field	Keyboard	
26	Combo box	Mouse	
27	Combo box	Mouse	
28	Date chooser	Mouse / Keyboard	
29	Label		
30	Layered pane		
31	Radio button	Mouse	Disables search text field and button & displays all records in table
32	Radio button	Mouse	Enables search text field and button
33	Radio button	Mouse	Enables search text field and button
34	Radio button	Mouse	Enables search text field and button
35	Radio button	Mouse	Enables search text field and button
36	Text field	Keyboard	
37	Button	Mouse	Searches/sorts the table records according to the radio button selected and data entered
38	Label		
39	Layered pane		
40	Label		
41	Label		
42	Layered pane		
43	Layered pane		
44	Button	Mouse	Activates and clears relevant data entry fields
45	Button	Mouse	Activates and inputs current data into relevant data entry fields
46	Button	Mouse	Clears data and deactivates data entry fields - displays current data
47	Button	Mouse	Executes SQL to add a new record to the table using entered data
48	Button	Mouse	Executes SQL to edit a record in the table using the edited data
49	Button	Mouse	Executes SQL to delete a record in the table
50	Button	Mouse	Leads to the Main Menu Frame

Visits Table



Visits Table Components

- **Level 1: Layered Pane - Cover whole screen to provide a border**
    - Label - Heading/Title label (Visits Table)
    - Button - Help button in top right
    - Button - Back button (leads back to LoginFrame)

- **Level 2: Layered Pane - Navigation (contains JTable)**
    - JTable - VisitsTable
    - Button - First record
    - Button - Previous record
    - Button - Next record
    - Button - Last record

- **Level 2: Layered Pane - Details**
    - Label - VisitNo
    - Text Field - VisitNo
    - Label - EventID
    - Text Field - EventID
    - Label - PatronID
    - Text Field - PatronID
    - Label - AmountSpent
    - Text Field - AmountSpent
    - Label - Username
    - Text Field - Username
    - Label - DateOfVisit
    - Date Chooser - DateOfVisit

- **Level 2: Layered Pane – Data Handling**
    - **Level 3: Layered Pane - Options**
        - Button - Add (allows data to be entered into the details pane for the add function)
        - Button - Edit (allows data to be edited in the details pane for the edit function)
        - Button - Delete (deletes the currently selected record)

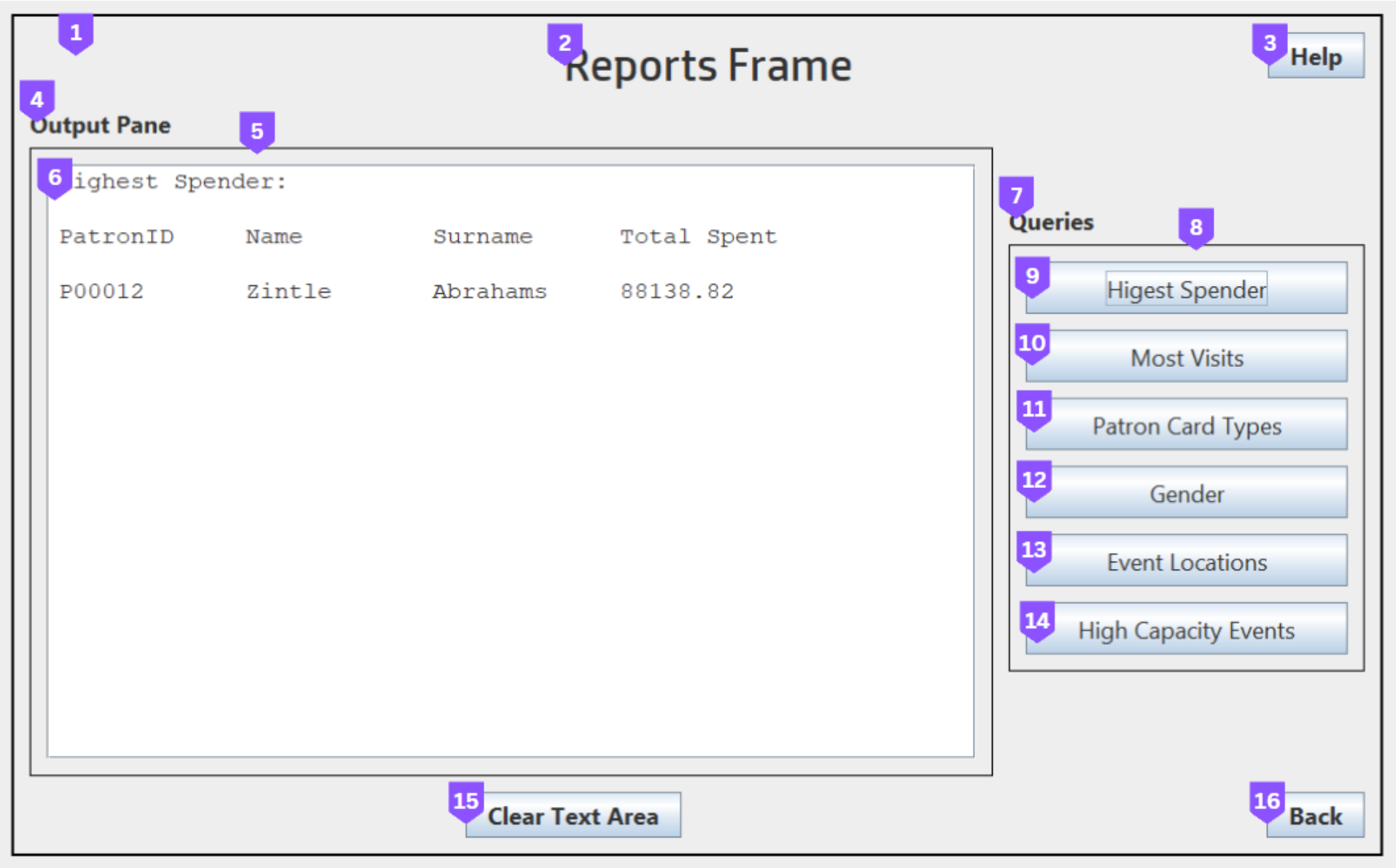
    - **Level 3: Layered Pane - Save Options**
        - Button - Save new (saves data entered in the details pane by calling the SQL method)
        - Button - Save edit (saves the data entered in the details pane by calling the SQL method)
        - Button - Cancel (makes the details pane inaccessible and removes all data entered/edited)

- **Level 2: Layered Pane - Search**
  - Radio Button - All (shows all records)
  - Radio Button - VisitNo (deactivates all other radio buttons and enables the text field for entry of a VisitNo)
  - Radio Button - PatronID (deactivates all other radio buttons and enables the text field for entry of a PatronID)
  - Radio Button - EventID (deactivates all other radio buttons and enables the text field for entry of a EventID)

NO	COMPONENT	INPUT	EVENT
0	Frame		
1	Layered pane		
2	Label		
3	Button	Mouse	Brings up the Help Frame
4	Label		
5	Layered pane		
6	Table		
7	Button	Mouse	Navigates to and displays the first record in the table
8	Button	Mouse	Navigates to and displays the next record in the table
9	Button	Mouse	Navigates to and displays the previous record in the table
10	Button	Mouse	Navigates to and displays the last record in the table
11	Label		
12	Layered pane		
13	Label		
14	Label		
15	Label		
16	Label		
17	Label		
18	Label		
19	Text field	Keyboard	
20	Check box	Mouse	Enables / Disables primary key field for manual editing
21	Combo box	Mouse	
22	Combo box	Mouse	
23	Text field	Keyboard	
24	Combo box	Mouse	
25	Date chooser	Mouse / Keyboard	
26	Label		

NO	COMPONENT	INPUT	EVENT
27	Layered pane		
28	Radio button	Mouse	Disables search text field and button & displays all records in table
29	Radio button	Mouse	Enables search text field and button
30	Radio button	Mouse	Enables search text field and button
31	Radio button	Mouse	Enables search text field and button
32	Radio button	Mouse	Enables search text field and button
33	Text field	Keyboard	
34	Button	Mouse	Searches/sorts the table records according to the radio button selected and data entered
35	Label		
36	Layered pane		
37	Label		
38	Label		
39	Layered pane		
40	Layered pane		
41	Button	Mouse	Activates and clears relevant data entry fields
42	Button	Mouse	Activates and inputs current data into relevant data entry fields
43	Button	Mouse	Clears data and deactivates data entry fields - displays current data
44	Button	Mouse	Executes SQL to add a new record to the table using entered data
45	Button	Mouse	Executes SQL to edit a record in the table using the edited data
46	Button	Mouse	Executes SQL to delete a record in the table
47	Button	Mouse	Leads to the Main Menu Frame

Reports Frame



Reports Frame Components

- **Level 1: Layered Pane – Background**
  - Label - Heading/Title label (Visits Table)
  - Button - Help button in top right
  - Button - Back button (leads back to LoginFrame)
  - Button - Clear Text Area

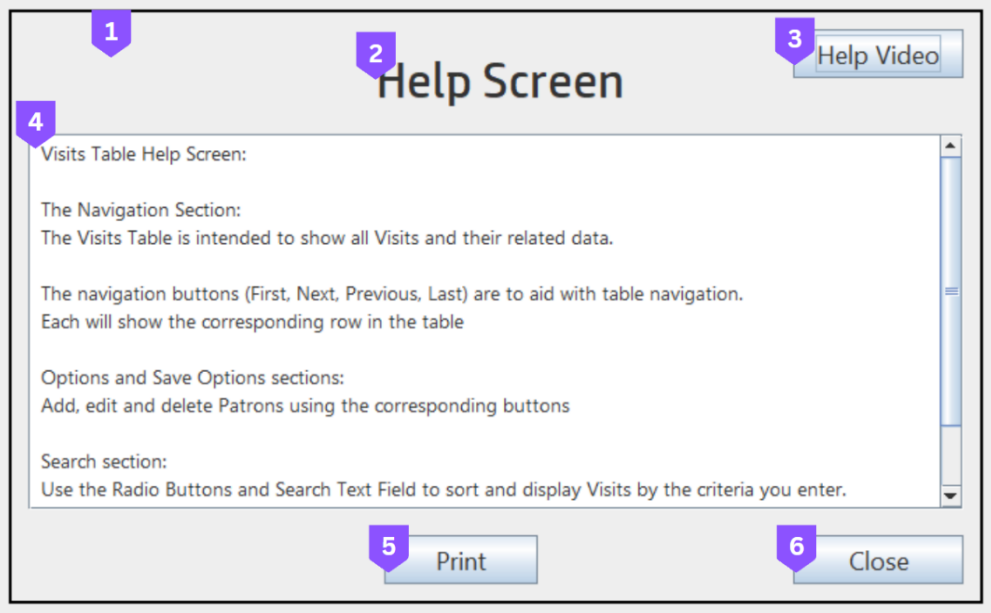
- **Level 2: Layered Pane - Output Pane (contains Text Area)**
  - Text Area - Display all queries when called

- **Level 2: Layered Pane - Queries (Contains buttons for each query)**
  - Button - Highest Spender
  - Button - Most Visits
  - Button - Patron Card Types
  - Button - Gender
  - Button - Event Location
  - Button - High-Capacity Events

NO	COMPONENT	INPUT	EVENT
0	Frame		
1	Layered pane		
2	Label		
3	Button	Mouse	Brings up the Help Frame
4	Label		
5	Layered pane		
6	Text area		
7	Label		
8	Layered pane		
9	Button	Mouse	Displays the result set for the Highest Spender query - The patron that spent the most
10	Button	Mouse	Displays the result set for the Most Visits query - The patrons that made the most visits
11	Button	Mouse	Displays the result set for the Patron Card Type query - The number of each card type
12	Button	Mouse	Displays the result set for the Gender query - The number of male and female patrons

13	Button	Mouse	Displays the result set for the Event Locations query - Number of events in each location
14	Button	Mouse	Displays the result set for the High Capacity Events query - Events with >1000 patrons
15	Button	Mouse	Clears the text area of all text
16	Button	Mouse	Leads to the Main Menu Frame

Help Frame



Help Frame Components

- Layered Pane – Background
  - Label – Heading
  - Button – Help
  - Button – Close
  - Button – Prints the contents of the text area
  - Text Area – Displays the help text from the help text files

NO	COMPONENT	INPUT	EVENT
0	Frame		
1	Layered pane		
2	Label		
3	Button	Mouse	Brings up the Help Frame
4	Text area		
5	Button	Mouse	Starts the printing process - brings up the printing configurations to begin printing
6	Button	Mouse	Closes the Help Frame

Exit Frame

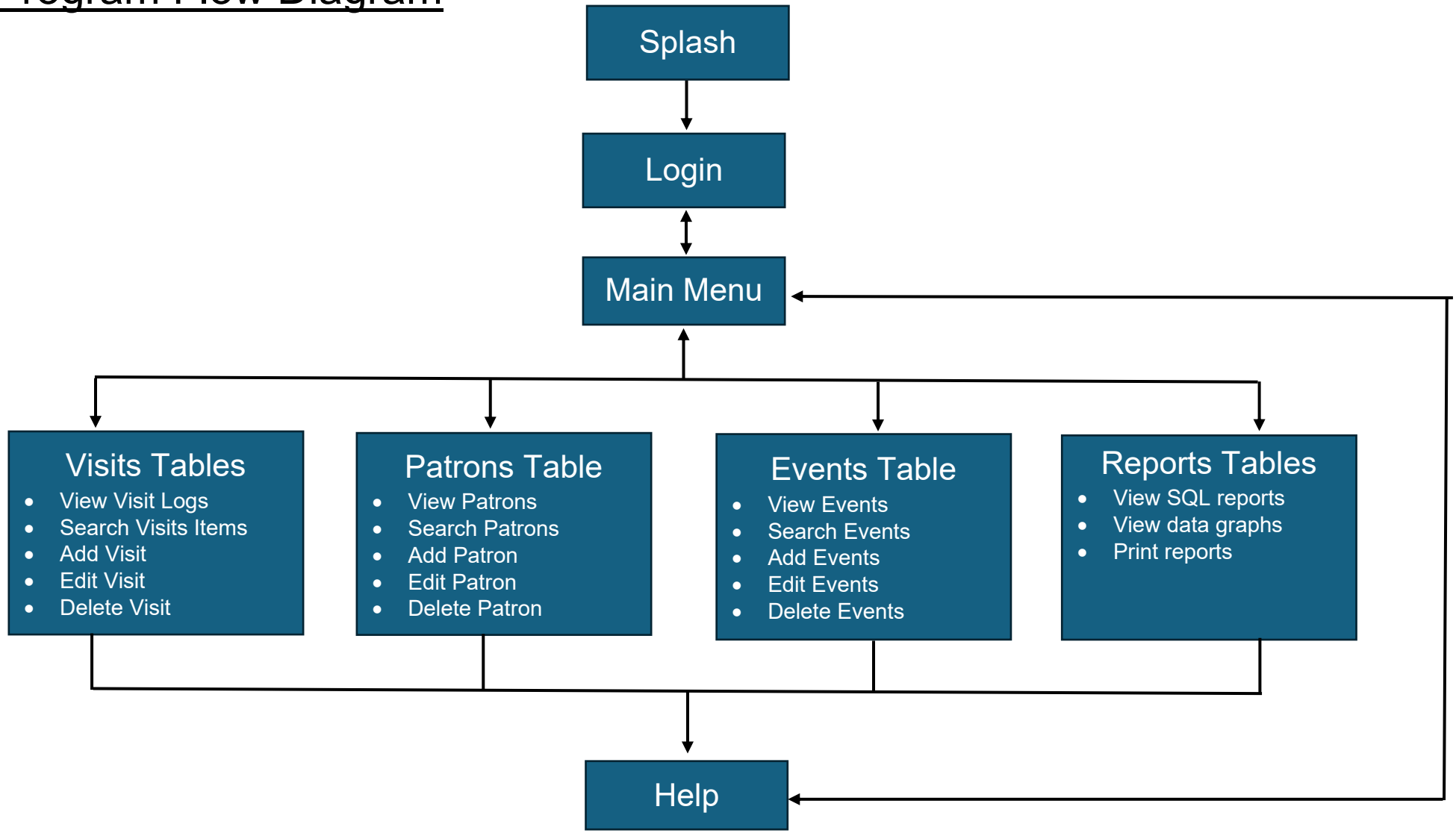
Exit Frame Components

- Layered Pane – Background
  - Label – Title(You are about to exit)
  - Label – Confirmation message(Are you sure you want to exit
  - Button – Exit(closes application)
  - Button – Return(goes back to LoginFrame)

NO	COMPONENT	INPUT	EVENT
0	Frame (not visible)		
1	Layered pane		
2	Label		
3	Label		
4	Button	Mouse	Leads to the Admin Login Frame
5	Button	Mouse	Closes the application

2.2 Program Flow Diagram

o



2.3 Class design AND OOP Principles(Class Diagrams)

Data classes

Users
- username : string - password : string - firstName : string - surname : string
+ Constructor() + Constructor(username : string, password : string, firstName : string, surname : string) + getUsername() : string + getPassword() : string + getFirstName() : string + getSurname() : string + setUsername(username : string) + setPassword(password : string) + setFirstName(firstName : string) + setSurname(surname : string) + toString() : string

Visits
- visitNo : integer - eventID : string - patronID : string - amountSpent : real - username : string - dateOfVisit : Date
+ Constructor() + Constructor(visitNo : integer, eventID : string, patronID : string, amountSpent : real, username : string, dateOfVisits : Date) + getVisitNo() : string + getEventID() : string + getPatronID() : string + getAmountSpent() : real + getUsername() : string + getDateOfVisit() : Date + setVisitNo(visitNo : string) + setEventID(eventID : string) + setPatronID(patronID : string) + setAmountSpent(amountSpent : real) + setUsername(username : string) + setDateOfVisit(dateOfVisit : Date) + toString() : string

Patrons
- patronID: string - firstName: string - surname: string - gender: string - dateOfBirth: Date - homeAddress: string - emailAddress: string - cardLevel: string - joinDate: Date
+ Constructor() + Constructor(patronID: string, firstName: string, surname: string, gender: string, dateOfBirth: Date, homeAddress: string, emailAddress: string, cardLevel: string, joinDate: Date) + getPatronID(): string + getFirstName(): string + getSurname(): string + getGender(): string + getDateOfBirth(): Date + getHomeAddress(): string + getEmailAddress(): string + getCardLevel(): string + getJoinDate(): Date + setPatronID(patronID: string) + setFirstName(firstName: string) + setSurname(surname: string) + setGender(gender: string) + setDateOfBirth(dateOfBirth: Date) + setHomeAddress(homeAddress: string) + setEmailAddress(emailAddress: string) + setCardLevel(cardLevel: string) + setJoinDate(joinDate: Date) + toString(): string

Events
- eventID: string - eventName: string - startDate: Date - endDate: Date - location: string - capacity: integer - status: string - registrationDeadline: Date
+ Constructor() + Constructor(eventID: string, eventName: string, startDate: Date, endDate: Date, location: string, capacity: integer, status: string, registrationDeadline: Date) + getEventID(): string + getEventName(): string + getStartDate(): Date + getEndDate(): Date + getLocation(): string + getCapacity(): integer + getStatus(): string + getRegistrationDeadline(): Date + setEventID(eventID: string) + setEventName(eventName: string) + setStartDate(startDate: Date) + setEndDate(endDate: Date) + setLocation(location: string) + setCapacity(capacity: integer) + setStatus(status: string) + setRegistrationDeadline(registrationDeadline: Date) + toString(): string

Object classes

UsersData
- db: DbManager - userList: List<Users> + Constructor() + getAllUsers(): + getAllUsers(username: string): + getAllUsers(userLevel: integer): + getUsersList(sql: string): + getUsername(username: string): Users - db: DbManager - userList: List<Users> + UsersData() + getAllUsers(): + getAllUsers(username: string): + getAllUsers(userLevel: integer): + getUsersList(sql: string):

VisitsData
- db: DbManager - ed: EventsData - pd: PatronsData - ud: UsersData - visitsList: List<Visits> + Constructor() throws SQLException + getAllVisits(): throws SQLException + getAllVisitsVisitNo(visitNo: integer): throws SQLException + getAllVisitsPatronID(patronID: string): throws SQLException + getAllVisitsEventID(eventID: string): throws SQLException + getAllVisitsUsername(username: string): throws SQLException + getVisitsList(sql: string): throws SQLException + getVisit(patronID: string): Visits + getVisitPosition(visitNo: integer): integer + addVisit(visitNo: integer, eventID: string, patronID: string, amountSpent: real, username: string, dateOfVisit: string): throws SQLException + editVisit(visitNo: integer, eventID: string, patronID: string, amountSpent: real, username: string, dateOfVisit: string): throws SQLException + deleteVisit(visitNo: integer): throws SQLException + populateEventIDJComboBox(eventIDCombo: javax.swing.JComboBox): + populatePatronIDJComboBox(patronIDCombo: javax.swing.JComboBox): + populateUsernameJComboBox(usernameCombo: javax.swing.JComboBox): + populateJTable(tblEvents: javax.swing.JTable, rowSelect: integer):

PatronsData
- db: DbManager - evd: EventsData - patronsList: List<Patrons> + Constructor() throws SQLException + getAllPatrons(): throws SQLException + getAllPatronsPatronID(patronID: string): throws SQLException + getAllPatronsFirstName(firstName: string): throws SQLException + getAllPatronsCardLevel(cardLevel: string): throws SQLException + getAllPatronsGender(gender: string): throws SQLException + getPatronList(sql: string): throws SQLException + getPatron(patronID: string): Patrons + getPatronPosition(firstName: string): integer + addPatron(patronID: string, firstName: string, surname: string, gender: string, dateOfBirth: string, homeAddress: string, emailAddress: string, cardLevel: string, joinDate: string): throws SQLException + editPatron(patronID: string, firstName: string, surname: string, gender: string, dateOfBirth: string, homeAddress: string, emailAddress: string, cardLevel: string, joinDate: string): throws SQLException + deletePatron(patronID: string): throws SQLException + populateJComboBoxCardLevel(patronCombo: javax.swing.JComboBox): + populateJComboBoxGender(patronCombo: javax.swing.JComboBox): + populateJTable(userTable: javax.swing.JTable, rowSelect: integer):

EventsData
- db: DbManager - eventsList: List<Events> - ev: EventsData + Constructor() throws SQLException + getAllEvents(): throws SQLException + getAllEventsEventID(eventID: string): throws SQLException + getAllEventsEventName(eventName: string): throws SQLException + getAllEventsLocation(location: string): throws SQLException + getEventsList(sql: string): throws SQLException + getEvent(eventID: string): Events + getEventPosition(eventName: string): integer + addEvent(eventID: string, eventName: string, startDate: string, endDate: string, location: string, capacity: integer, status: string, registrationDeadline: string): throws SQLException + editEvent(eventID: string, eventName: string, startDate: string, endDate: string, location: string, capacity: integer, status: string, registrationDeadline: string): throws SQLException + deleteEvent(eventID: string): throws SQLException + populateLocationJComboBox(locationCombo: javax.swing.JComboBox): + populateStatusJComboBox(statusCombo: javax.swing.JComboBox): + populateJTable(tblEvents: javax.swing.JTable, rowSelect: integer):

DbManager
conn: Connection + Constructor() + query(SQL: string): ResultSet throws SQLException + update(SQL: string): integer throws SQLException + updateReturnID(SQL: string): integer throws SQLException

ReportsData
db: DbManager + Constructor() throws SQLException + pieChart(): throws SQLException + barGraph(): throws SQLException + getHighestSpender(): String throws SQLException + getMostVisitsPatron(): String throws SQLException + getEventLocation(): String throws SQLException + getPatronCardType(): String throws SQLException + getPatronGender(): String throws SQLException + getHighCapacityEvents(): String throws SQLException

Validation Class

Validation
db: DbManager
+ vVisitNoCheck(valueToCheck: string): real
+ vPKCheck(valueToCheck: string, initial: char, primaryKeyField: string, table: string): real
+ vAgeMin(dob: Date, minAge: integer): real
+ vDateMin(endDate: Date, startDate: Date): real
+ vDateMax(entry: Date, startDate: Date): real
+ vBasicPresenceCheck(field: string): real
+ v2IntialCharacters(numChars: integer, condition: string, entry: string): real
+ vVirtualCombo(entry: string, option1: string, option2: string): real
+ vVirtualCombo3(entry: string, option1: string, option2: string, option3: string): real
+ vVirtualCombo4(entry: string, option1: string, option2: string, option3: string, option4: string): real

2.4 Secondary Storage Design

Database will require the following tables containing appropriate data with the following structure. Sample data is also provided.

TblUsers

Design View

TblUsers

Field Name	Data Type	Description (Optional)
Username	Short Text	The primay key, unique identifier for each admin user that oversees the database. Admins log visits and will be seen in TblVisits
Password	Short Text	Password that corresponds with the username above. Used for logging in to the application
FirstName	Short Text	The first name of the admin user
Surname	Short Text	The surname of the admin user

Datasheet view

TblUsers

Username	Password	FirstName	Surname	Click to Add
U1001	Admin1	Vashni	Naidoo	
U1002	Admin2	Satoru	Gojo	
U1003	Admin3	Suguru	Geto	
U1004	Admin4	Miwa	Miyawaki	
U1005	Admin5	Jody	Quartz	
U1006	Admin6	Lika	Govender	
U1007	Admin7	Wyatt	Chinner	

Description

TblUsers stores information about the admin users which oversee the database. The username and password data will be used for logging in to the application upon startup. These admins also make visit logs found in TblVisits. As a result, their username will appear in TblVisits next to the data of the visit they have logged. This is to ensure accountability and easy tracing of logs.

TblVisits

Design View

TblVisits

Field Name	Data Type	Description (Optional)
VisitNo	Number	Primay key, visit number assigned to each visit for identification purposes
EventID	Short Text	The primary key of TblEvents. Used to uniquely identify events
PatronID	Short Text	The primary key of TblPatrons. Used to uniquley identify patrons
AmountSpent	Number	The amount of money spent during this visit
Username	Short Text	The admin user that recordeed this visit
DateOfVisit	Date/Time	The date of the visit

TblVisits

VisitNo	EventID	PatronID	AmountSpe	Username	DateOfVisit	Click to Add
1	E00009	P00015	150,00	U1006	2023/10/25	
2	E00010	P00018	45454,00	U1006	2024/07/07	
3	E00001	P00001	4874,00	U1001	2024/07/09	
4	E00008	P00012	88138,82	U1002	2023/10/16	
5	E00011	P00011	76087,49	U1002	2023/04/10	
6	E00009	P00015	343,00	U1006	2024/07/07	
7	E00010	P00013	38443,07	U1002	2023/10/16	
8	E00003	P00002	58917,77	U1002	2023/10/17	
9	E00012	P00015	21876,91	U1002	2023/10/18	
10	E00005	P00011	100,00	U1002	2023/10/25	

Datasheet View

Description

TblVisits is a linking table that makes use of data from all other tables. It acts as a log book that records any and all visits to the casino made by patrons as well as data related to the activity of the patron during the visit

TblEvents

Design View

TblEvents

Field Name	Data Type	Description (Optional)
EventID	Short Text	Primary key, unique code used to identify each event
EventName	Short Text	Name of the event
StartDate	Date/Time	The date the event starts and can be participated in
EndDate	Date/Time	The date the event ends and can no longer be participated in
Location	Short Text	Events can be held either online or on premises
Capacity	Number	Maximum amount of people that can attend/participate
Status	Short Text	The status of the event: Scheduled, cancelled or completed
RegistrationDeadline	Date/Time	The latest time that a patron may register for an event, These are one week before the start date usually

TblEvents

EventID	EventName	StartDate	EndDate	Location	Capacity	Status	RegistrationDeadline	Click to Add
E00001	Summer Smash	2024/07/12	2024/07/14	Physical Venue	1500	Scheduled	2024/07/04	
E00002	Festival Des Flores	2024/07/08	2024/07/08	Physical Venue	2000	Scheduled	2024/07/07	
E00003	Neon Nights	2024/07/01	2024/07/04	Physical Venue	1501	Scheduled	2024/07/05	
E00004	Winning Wednesday	2023/09/02	2023/09/28	Online	2000	Scheduled	2023/08/26	
E00005	Retro Revival	2023/07/08	2023/07/10	Online	2000	Scheduled	2023/07/01	
E00006	Imposters	2023/10/08	2023/10/24	Online	1000	Scheduled	2023/10/27	
E00007	Mystery Mansion Escape	2023/08/14	2023/09/05	Physical Venue	750	Scheduled	2023/08/07	
E00008	Radiant Reels	2023/05/05	2023/05/06	Physical Venue	950	Scheduled	2023/04/28	
E00009	Roulette Rendezvous	2023/09/20	2023/10/15	Physical Venue	870	Scheduled	2023/09/13	
E00010	Twilight Table	2023/04/07	2023/04/08	Online	2000	Scheduled	2023/03/31	

Datasheet View

Description

TblEvents stores data about all the events of the casino, active, inactive, upcoming, cancelled and so on. All events are stored for recording purposes.

TblPatrons

Design View

TblPatrons

Field Name	Data Type	Description (Optional)
PatronID	Short Text	The primary key, unique code used to identify each patron
FirstName	Short Text	The first name of the patron
Surname	Short Text	The surname of the patron
Gender	Short Text	The gender of the patron: Male, Female or Other
DateOfBirth	Date/Time	The date the patron was born
HomeAddress	Short Text	Patron's place of residence
EmailAddress	Short Text	Patron's email address
CardLevel	Short Text	Type of card: Black, Silver, Gold, or Platinum
JoinDate	Date/Time	The date that the patron registered and was added to the database

Datasheet view

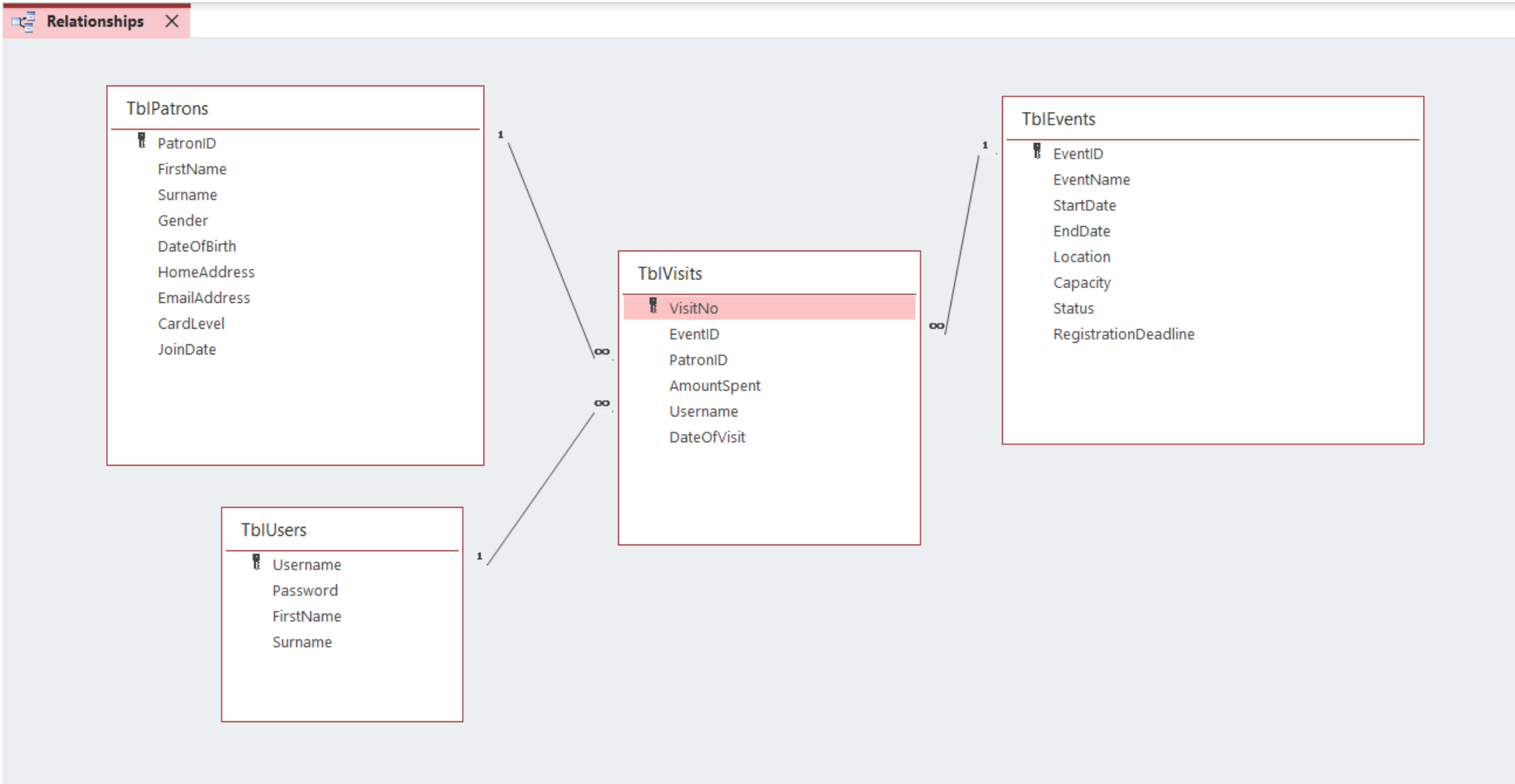
TblPatrons

PatronID	FirstName	Surname	Gender	DateOfBirth	HomeAddress	EmailAddress	CardLevel	JoinDate	Click to Add
P00001	Eren	Yeager	Male	2000/10/30	Shiganshina District, Wall Sina	foundingtitan@gmail.com	Silver	2024/07/03	
P00002	David	Smith	Male	1990/03/15	123 Main St, Pretoria, 0182	david.smith_1987@gmail.com	Silver	2021/01/15	
P00003	Carla	Jansen	Female	2005/06/28	456 Elm Ave, Bloemfontein, 9302	carla_jansen123@yahoo.com	Gold	2021/07/27	
P00004	Sai	Mungroo	Male	2023/10/01	Mooi River, 1348	sai@gmail.com	Platinum	2023/10/25	
P00005	Thando	Botha	Male	1992/09/20	789 Oak Rd, Kimberley, 8302	thando.botha_456@gmail.com	Silver	2020/12/10	
P00006	Michael	Van Der Walt	Male	2000/02/12	101 Pine Ln, Polokwane, 0701	michael.walt@yahoo.com	Platinum	2020/05/05	
P00007	Fatima	Jacobs	Female	1994/11/05	222 Cedar Rd, Nelspruit, 1201	fatima.jacobs_1990@gmail.com	Silver	2021/03/18	
P00008	Grant	Brown	Male	2002/07/18	333 Elm St, East London, 5202	grant_brown456@yahoo.com	Gold	2020/09/03	
P00009	Lerato	Sibanda	Female	1995/04/10	444 Redwood Ave, Pietermaritzburg, 3202	lerato_sibanda_789@gmail.com	Black	2021/02/14	
P00010	Jessica	Adams	Female	1993/12/22	555 Birch Ave, George, 6530	jessica.adams456@yahoo.com	Gold	2020/11/22	

Description

TblPatrons is a contact list type of table that holds all the patrons and their related information.

Relationships Diagram



Help Textfiles

Admin Login help Screen

AdminLogin - Notepad

File Edit Format View Help

```
Admin Login Help Screen:

Please enter your User PIN and Password to continue to the Table Selection screen

Username Example: U1005
Password Example: Admin5
```

Ln 1, Col 1 | 100% | Windows (CRLF) | UTF-8

Table Help Screen (.txt file content will differ for each table)

PatronsTable - Notepad

File Edit Format View Help

```
Patrons Table Help Screen:

The Navigation Section:
The Patrons Table is intended to show all Patrons and their related data.

The navigation buttons (First, Next, Previous, Last) are to aid with table navigation.
Each will show the corresponding row in the table

Options and Save Options sections:
Add, edit and delete Patrons using the corresponding buttons

Search section:
Use the Radio Buttons and Search Text Field to sort and display Patrons by the criteria you enter.

Please enter valid data
```

Ln 16, Col 1 | 100% | Windows (CRLF) | UTF-8

Reports Frame Help Screen

ReportsFrame.txt - Notepad

File Edit Format View Help

```
Reports Frame Help Screen:

Please click on a button to view the corresponding information relating to the various tables

All data should be displayed in the Text Area
Graphs serve to aid in data visualisation and understanding
```

Ln 6, Col 17 | 100% | Windows (CRLF) | UTF-8

2.5 Explanation of Secondary Storage

Why Choose a Database Over JSON or Text Files

Data Security, Integrity, and Consistency

Databases: Databases ensure data integrity and consistency through constraints (e.g., primary keys, foreign keys), validation techniques and so on. The ability to make sure data is valid is a huge concern. Data is also safer due to the password protection and other safety features present in database software.

JSON/Text Files: JSON or text files lack built-in mechanisms to enforce data integrity and consistency. Working with data without these mechanisms is error-prone and inefficient. JSON/Textfiles are largery unsafe due to their inablity to use password protection and other security features.

Complex Queries and Relationships

Databases: Databases excel at handling complex queries and relationships between entities. SQL allows for sophisticated querying and reporting, which is essential for generating insights from my casino data (e.g., patron visit history, event participation).

JSON/Text Files: Performing complex queries on JSON or text files is cumbersome and slow. You would need to load and parse entire files, which is inefficient for large datasets.

Scalability and Performance

Database: Databases are optimized for performance and can handle large volumes of data efficiently. Indexes, query optimization, and caching improve data retrieval times.

JSON/Text Files: As the data grows, the performance of JSON or text files degrades significantly. File I/O operations are slower, and parsing large files in memory can be resource-intensive.

Backup and Recovery

Databases: Databases provide robust backup and recovery options, including automated backups, point-in-time recovery, and replication.

JSON/Text Files: Implementing reliable backup and recovery processes for JSON or text files requires additional effort and custom solutions.

Implications of Using a Database

Structured Data Storage

My design makes use of relational database tables (TblUsers, TblVisits, TblEvents, TblPatrons) to store date in a structured and efficient manner. Each table has clearly defined fields and data types, ensuring data is stored in a consistent format.

Data Relationships

The use of foreign keys (e.g., EventID in TblVisits referencing TblEvents, PatronID in TblVisits referencing TblPatrons) allows for establishing relationships between tables. This supports complex queries to retrieve related data across multiple tables, such as finding all visits made by a particular patron to specific events.

Data Integrity

Primary keys (e.g., Username in TblUsers, VisitNo in TblVisits, EventID in TblEvents, PatronID in TblPatrons) ensure each record is uniquely identifiable, preventing duplication and maintaining data integrity.

Efficient Data Retrieval

Indexes can be created on frequently queried fields (e.g., Username, EventID, PatronID) to speed up data retrieval operations, improving the overall performance of your casino management system.

Scalability

As your casino grows and the volume of data increases, the database can scale to accommodate the growth. Advanced database features like partitioning, indexing, and replication support scalability and high availability.

Security

Databases provide robust security features, including user authentication, role-based access control, and encryption. This is vital for protecting sensitive information about patrons, events, and visits.

2.6 Explanation of how Primary Data Structures relate to Secondary Storage

For every table that is present in secondary storage, there is a corresponding Object Class in the primary storage design. The object class corresponds with the table using different fields/class variables to represent each column in the table, each field being of the same data type as its corresponding column. A simple object in the object class will have data that relates to one record (row) in its corresponding table.

Each of these primary storage object classes has an associated data class. A data class refers to a class in which there is a list. This list stores many objects from one object class and thus stores multiple records from a table in the database. Auxiliary methods are used to process data in secondary storage and allow this data/each record to be stored in a list.

This relationship between secondary storage, primary storage and the associated data class is explained further through the following table and paragraph points:

Primary Storage		Secondary Storage
Object class	Data Class	Database Table
Users	UsersData	TblUsers
Events	EventsData	TblEvents
Patrons	PatronsData	TblPatrons
Visits	VisitsData	TblVisits

Users:
The primary function of users is to facilitate login. The Users class stores the attributes. The methods in the UsersData class are primarily for checking the credentials against the database to facilitate the login procedure. The method from the UsersData class is called by in the login button in the Admin Login screen
Data is pulled from the database when login credentials need to be verified.

Events:
The primary function of events is to keep a list of all events. The Events class stores the attributes. The methods in the EventsData class are primarily methods that are used in the GUI frame for displaying, searching/sorting and table functions.

Patrons:
The primary function of patrons is to keep a list of all patrons. The Patrons class stores the attributes. The methods in the PatronsData class are primarily methods that are used in the GUI frame for displaying, searching/sorting and table functions.

Visits:
The primary function of visits is to keep a list of all casino visits. The Visit class stores the attributes. The methods in the VisitsData class are primarily methods that are used in the GUI frame for displaying, searching/sorting and table functions.