ESOF 322: Project 1

Jake Coleman, William Jardee, Fletcher Philips, Megan Steinmasel

November 9, 2022

General Feature:

User Story 1: As a website user, I want a functional navigation menu and search bar so that I can access information.

User Story 3: As a website user, I need the processing of new data to find important features and calculate relevant statistics to be done automatically, so it is friendly to someone that is not a data scientist.

User Story 4: As a website user, I want a data visualization technique that is intuitive and accurately shows patterns in the data.

User Story 5: As a database engineer, I want to have a database so that I can store cold cases inside of it.

User Story 6: As a database engineer, I want the data to be manipulatable so that the admin can insert and delete data from the database.

Author: Jake Coleman

Name:	Visualize Results
Description:	Displays selected graphs from the gathered data
Related Requirements:	User Story 2
Preconditions:	The website user has signed into the website and accessed the drop-down menu to access the page. Some data has been selected for an analysis and the desired type of graph has been selected.
Successful end condition:	The graphs are displayed on a separate page to view.
Failed end condition:	Fails to display any graphs
Actors:	Website User
Basic Flow of Events:	1. The website user goes through the drop-down menu process 2. The website user chooses to view graphs via drop-down menu 3. The system will display any applicable graphs
Extensions/Exceptional Flow of Events	1. The graphs fail to display 2. the User is notified that the use is unable to access graphs via a notification 3. write to log file 4. User is sent back to main navigation system

Author: Fletcher Philips

Name:	Create Database
Description:	A database needs to be created and connected to the web framework we choose
Related Requirements:	User Story 5
Preconditions:	Basic web infrastructure has been created. Small sample set of cold cases is ready to be inserted.
Successful end condition:	Basic database has been initialized with substructure. Data has been inserted into the database.
Failed end condition:	Data cannot be inserted into the database.
Actors:	Database engineer
Basic Flow of Events:	1. Create MySQL Database. 2. Connect a database to the web framework 3. Insert Data into MySQL database
Extensions/Exceptional Flow of Events	1. Database fails to build properly 2. the engineer is notified that an error has appeared 3. write to log file 4. engineer is ejected from the system

Author:

Name:	
Description:	
Related Requirements:	
Preconditions:	
Successful end condition:	
Failed end condition:	
Actors:	
Basic Flow of Events:	1. .
Extensions/Exceptional Flow of Events	1. .

Author: William Jardee

Name:	Manage Data
Description:	Data needs to be inserted, deleted, and manipulated. Related to this, there must be an appropriate Graphical User Interface. The data will connect directly with the database.
Related Requirements:	User Story 3, User Story 6
Preconditions:	The user is logged on and has gained access rights according to their credentials.
Successful end condition:	Data is successfully manipulated and success code received from source.
Failed end condition:	Requested task is outside of credentials. Success code not received. Invalid new data
Actors:	Website User, Database engineer
Basic Flow of Events:	1. Actor selects the action they wish to commit, and what to commit it on. 2. Action is tested against credentials 3. Success/Fail state is determined 4. any follow-up effects happen (i.e., Visualize Results Via Graph) 5. Flow is complete and prompts user for next action
Extensions/Exceptional Flow of Events	1. Conflict happens 2. Reject any attempted changes and revert to the last viable state 3. Notify actor that there has been an error and write to log file 4. Flow is complete and prompts user for next action

Author: William Jardee

Name:	Create Log
Description:	A log file should be kept to keep track of flow as to diagnose errors and suspicious behavior.
Related Requirements:	Catch all location for all errors
Preconditions:	The system has been started effectively and there is a safe place to store a text file (log file).
Successful end condition:	Data can be saved to the log file
Failed end condition:	Data cannot be safely save to log file
Actors:	Database engineer
Basic Flow of Events:	1. Write to file recent activity 2. Flag any invalid actions that prompt “write to log file”
Extensions/Exceptional Flow of Events	1. Notify system admin of issue and include error information 2. Terminate all systems until issue is resolved

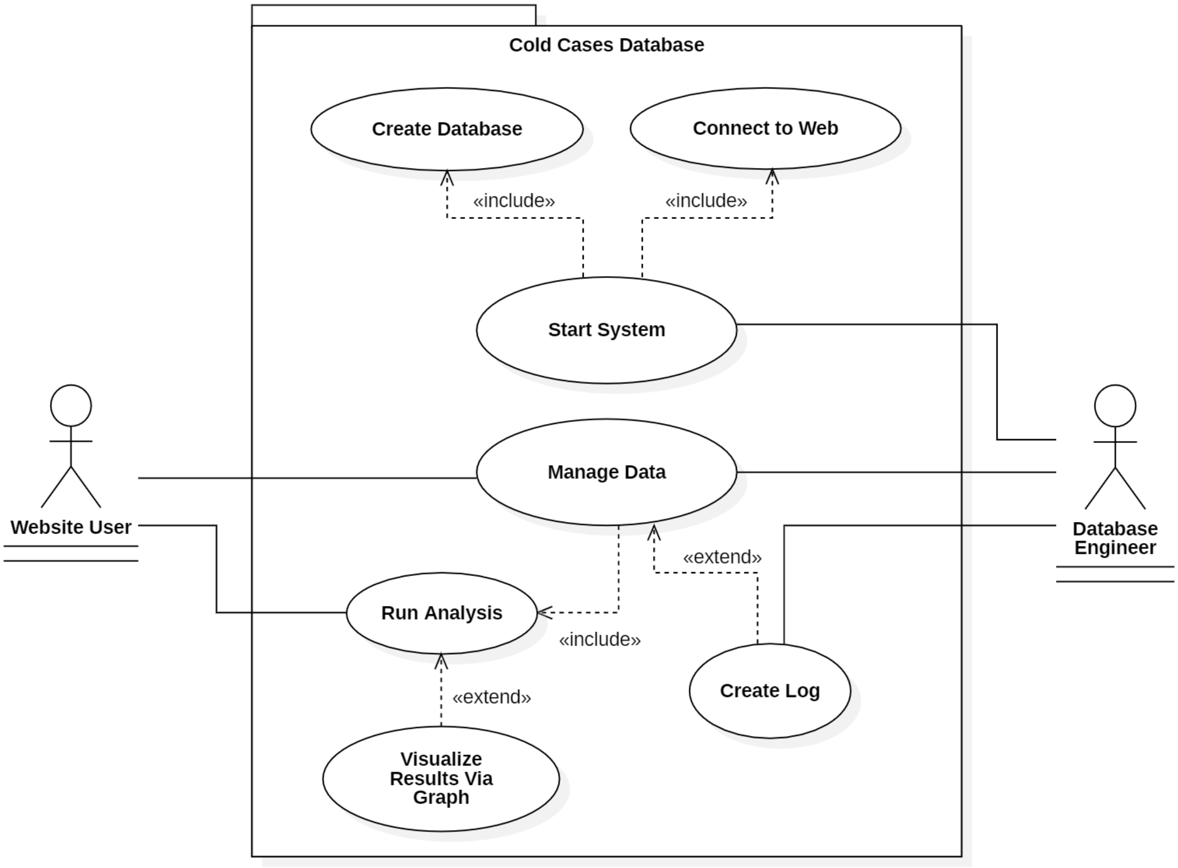

Figure 1: UseCase diagram for the Cold-Cases Database system.

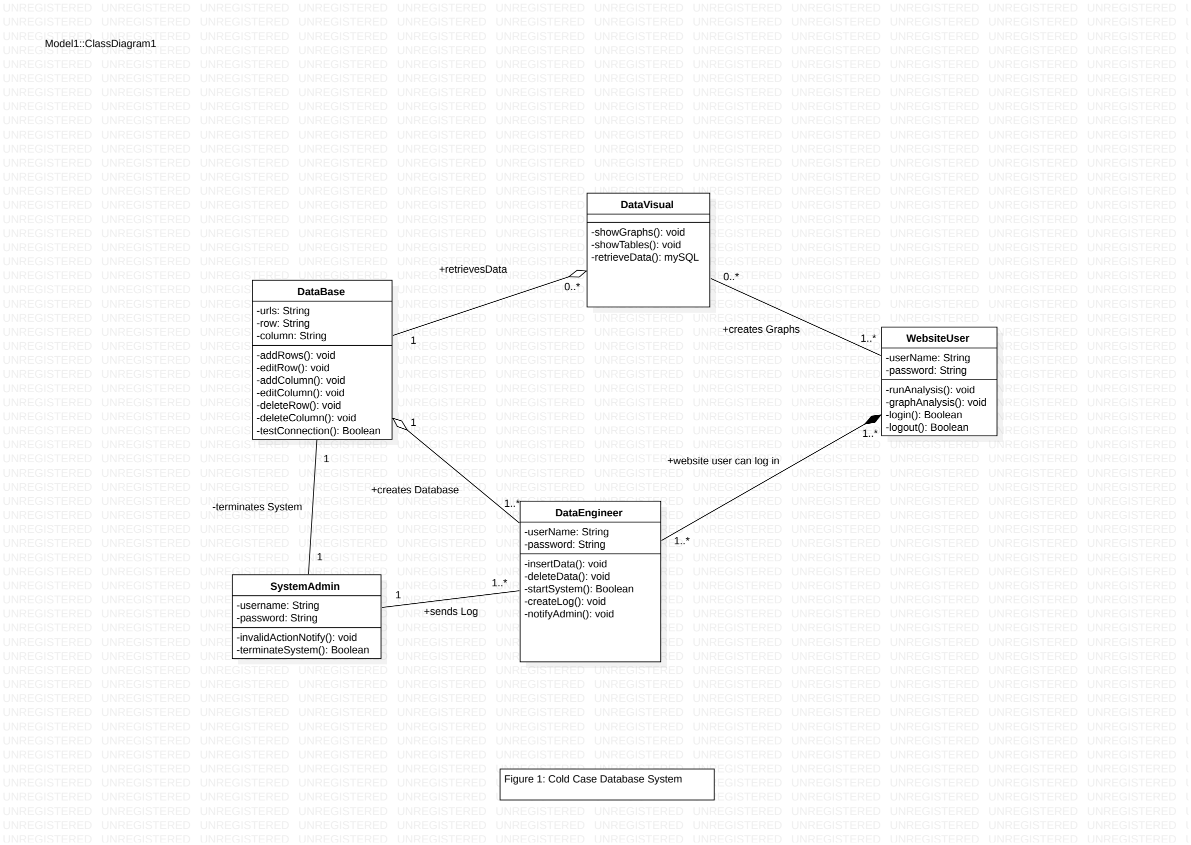

Figure 2: Class diagram for the Cold-Cases Database system.

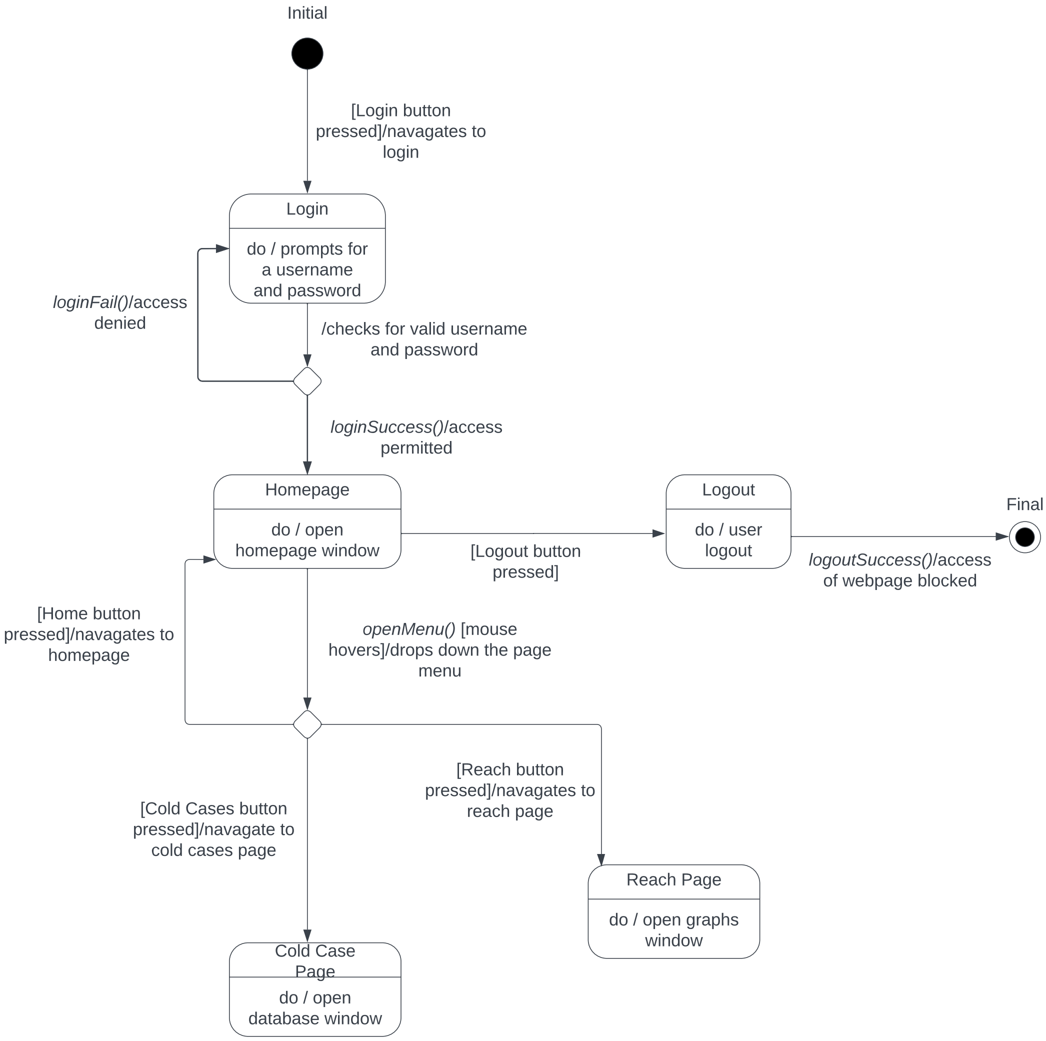
Initial
[Login button pressed]/navagates to login
Login
do / prompts for a username and password
loginFail()/access denied
/checks for valid username and password
loginSuccess()/access permitted
Homepage
do / open homepage window
[Logout button pressed]
Logout
do / user logout
logoutSuccess()/access of webpage blocked
Final
[Home button pressed]/navagates to homepage
openMenu() [mouse hovers]/drops down the page menu
[Reach button pressed]/navagates to reach page
[Cold Cases button pressed]/navagate to cold cases page
Reach Page
do / open graphs window
Cold Case Page
do / open database window

Figure 3: State Chart diagram for the Cold-Cases Database system.

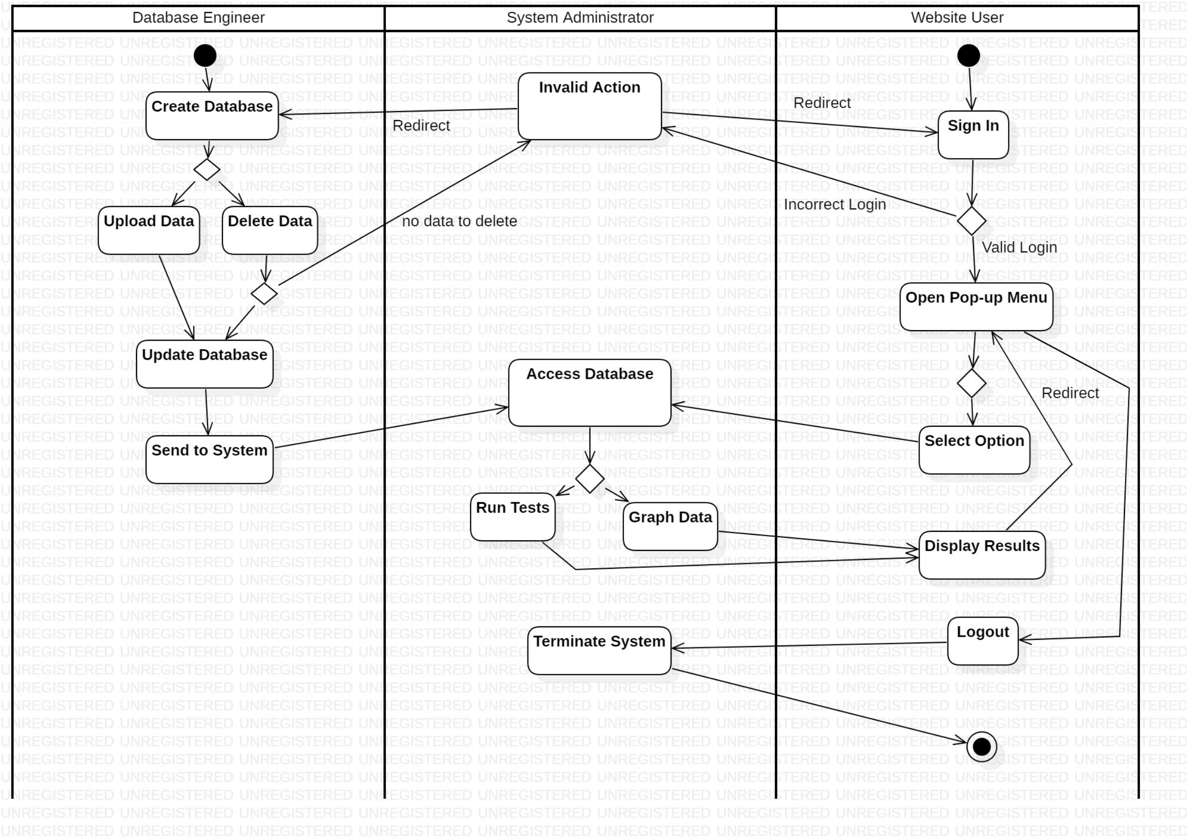

Figure 4: Activity diagram for the Cold-Cases Database system.