Faculty of ICT, Mahidol University

Project: Phase 3

By

Group: _init_.py

6488004 Kittipich Aiumbhornsin

6488073 Chalisa Sae-ngow

6488087 Thanatchaya Sujarit

6488089 Pattaravit Suksri

6488134 Jirateep Rudeerudchanawong

6488168 Linfeng Zhang

Submitted to

ITCS371 Software Engineering

Asst. Prof. Dr. Thanwadee Sunetnanta

Asst. Prof. Dr. Morakot Choetkiertikul

Dr. Chaiyong Ragkhitwetsagul

A report submitted

as the partial fulfillment of the requirements for the project

November 2023

Requirements & Constraints

Following tables are the requirements which are divided into requirements categories: functional requirements and non-functional requirements. The code of the requirements will be in the *“FRXX”* format and *“NRXX”* format where *“FR”* stands for functional requirements and *“NR”* stands for non-functional requirements while *“XX”* is the number of particular requirements. Similarly, for the constraints, *“CTXX”* format will be used where *“CT”* stands for constraints and *“XX”* is the number of particular constraints.

Functional Requirements

Code	Requirements	Importance
FR01	Payment can only be done using credit cards.	Must
FR02	Users are able to search for the game.	Must
FR03	System must provide a wishlist system for each user where users can modify their list at any time.	Must
FR04	Users are able to evaluate the game they have played on the scale of 1 to 5 stars.	Must
FR05	System is able to list all of the games that are available on the system to users.	Must
FR06	Users are able to choose the kind or genre of the game they would like to play.	Must
FR07	Each game in the system must provide the specification of the machine that the game can be played on.	Must
FR08	System is able to tell the price and appropriateness rating of the game.	Must
FR09	Administrator of the system can manage and organize the game list in the system.	Must
FR10	Game creators must be able to upload game executables, provide descriptions, and set prices for their games.	Must
FR11	Users must be able to create and manage their accounts.	Must
FR12	Users must have an option for downloading the game to their computer or store in the cloud.	Must
FR13	Must have items market to sell items in the game to other users.	Must
FR14	User data and credit card information must be encrypted for security.	Must

FR15	Games must be available to download through the system.	Must
FR16	Users are able to obtain digital licenses of the game by purchasing the game through the system.	Must
FR17	Applications must have a shopping cart for multiple games in the cart and be paid at once.	Must
FR18	Able to manage digital rights of the game.	Must
FR19	User information must have a name, address and credit card.	Must
FR20	Users are able to create an account.	Must
FR21	System should be able to announce the upcoming available games to users at some part of the user interface.	Should

Non-Functional Requirements

Code	**Requirements**	**Importance**
NR01	The system must have 99.999% uptime guarantee.	Must
NR02	Have a nice looking user interface with good user experience design.	Must
NR03	Game data is saved in the cloud.	Must
NR04	Ensure the security of the data.	Must
NR05	Data transfer between the system and users must be encrypted.	Must
NR06	System must be able to handle 10,000 active users at the same time.	Must
NR07	The application must be natively compatible with Windows-based PCs.	Must
NR08	Cloud storages must be available for users to store their games.	Must
NR09	User interface must be attractive to users.	Must
NR10	The response time of the system should be no longer than 1 second for all user interactions.	Should

Constraints

Code	Constraints	Importance
CT01	System is only available on Windows computers only.	Must
CT02	The games must be age-restricted.	Must

Previous phase revision

Revision Log from Phrase2

Added

1. Account Manager: Responsible for managing user details, enabling efficient user profile administration.
2. Game Library: Offers a comprehensive overview of game statuses, allowing users to keep track of their gaming experiences.
3. Wish List: Empowers users to curate and manage their gaming aspirations.
4. Cloud Data Manager: Seamlessly handles game data and cloud storage, safeguarding essential information and ensuring easy accessibility.
5. Search System: Simplifies game discovery with robust keyword-based search capabilities.
6. Game Rating System: Allows users to rate and share feedback on games they've played.
7. Admin System: Approves or declines new games, maintaining quality control.
8. Creator System: Supports game creators, providing a platform for developers to showcase their work and interact with the community.
9. Market System: Lists games for potential buyers and offers developers a platform to market their creations.
10. Two-Factor Authentication & Notification System: Enhances user security and settings with two-factor authentication and keeps users informed through notifications.
11. Sign Up & Authentication System: Streamlines user registration and ensures secure authentication, providing access to the platform's services.
12. Cart System: Enables seamless payment processing, ensuring a smooth user experience when purchasing games and in-game items.

Corrected

1. Data can’t flow between GameData and User (both are entities)

Revision Log Phase 3

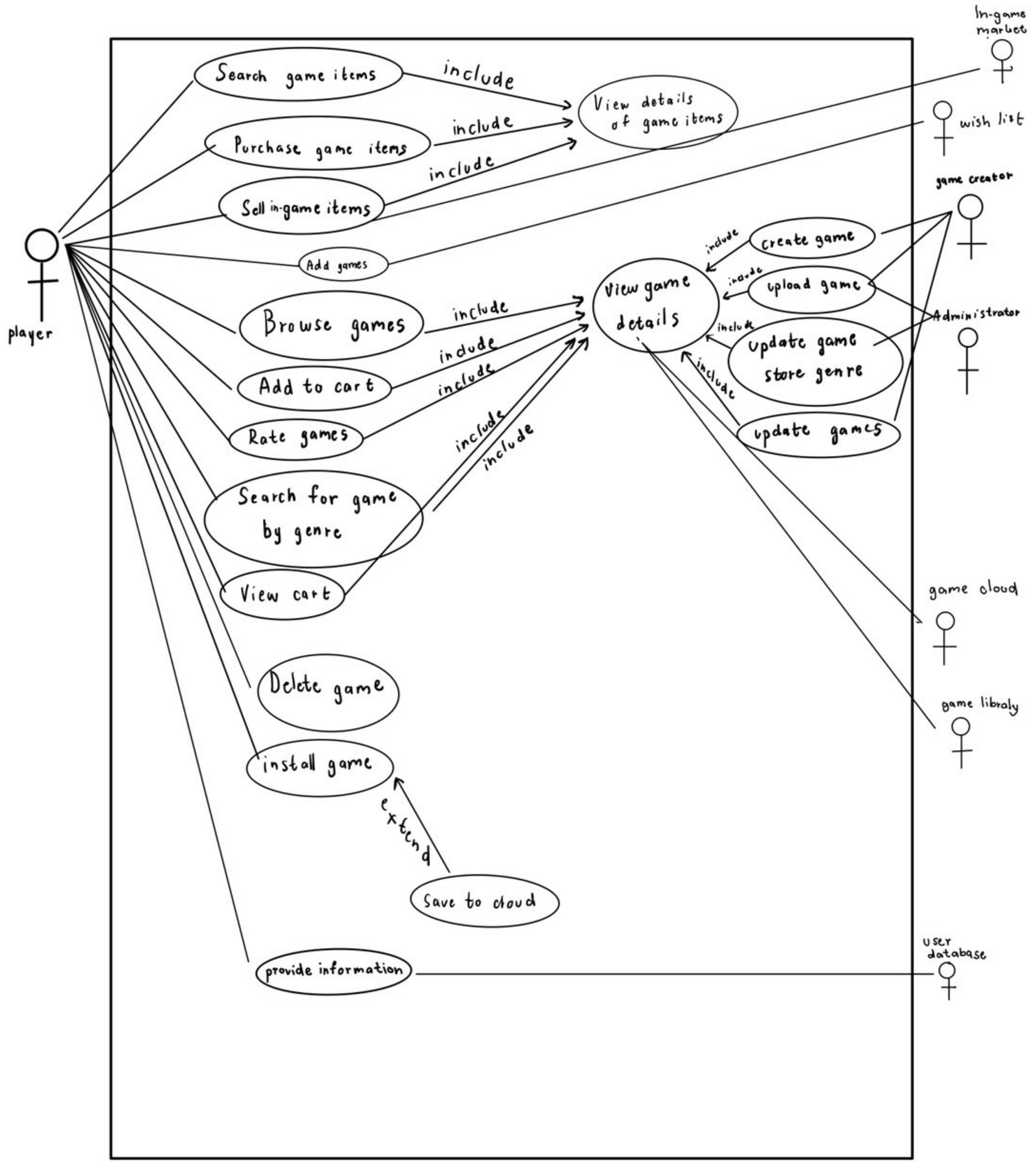
Change

1. Add games —-> purchase
2.provide information —-> log in

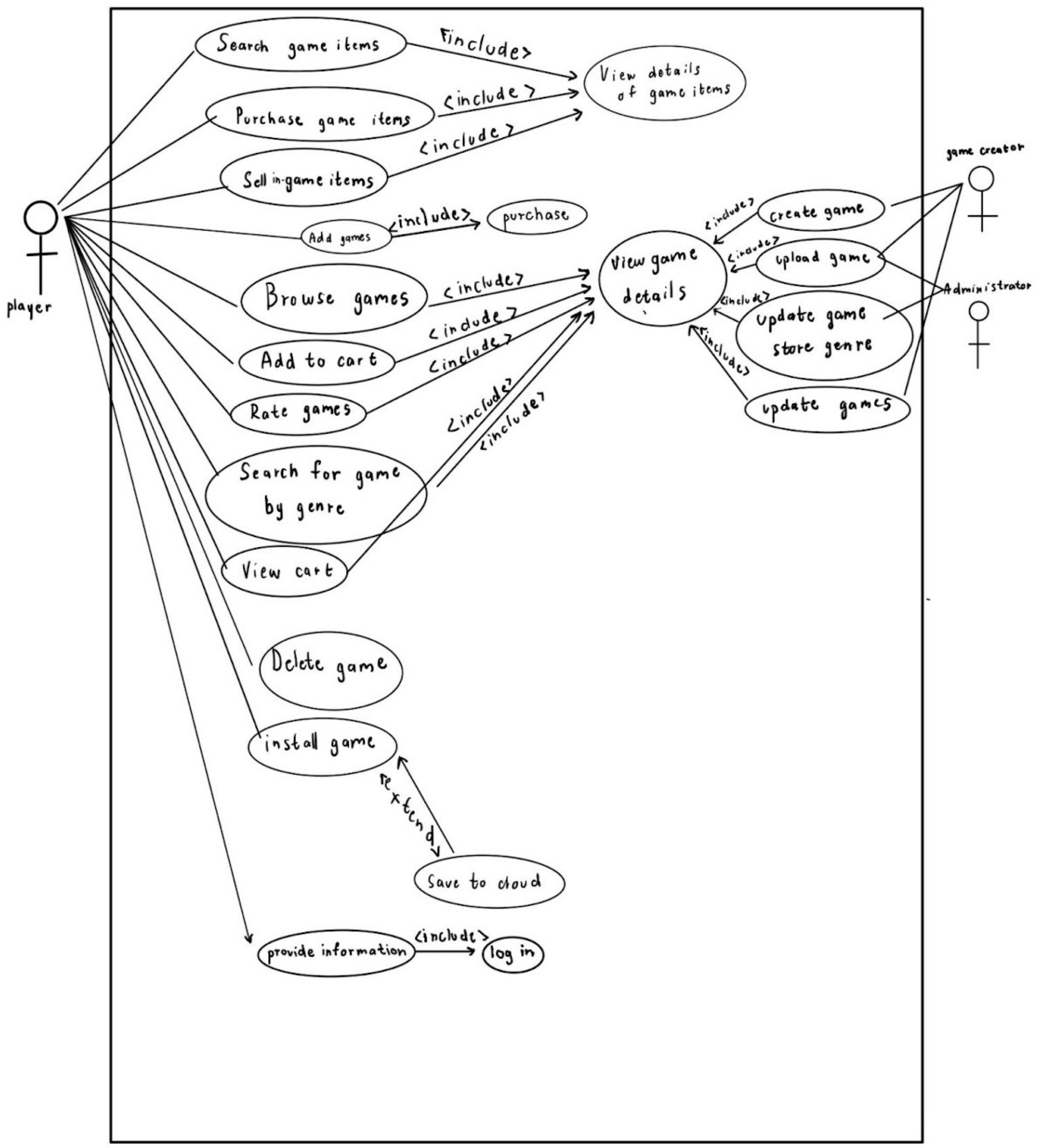
Corrected

1.Delete In-game market because in use case does not need to show (for user)
2.Delete Wish list because in use case does not need to show (for user)
3.Delete User Database because in use case does not need to show (for user)

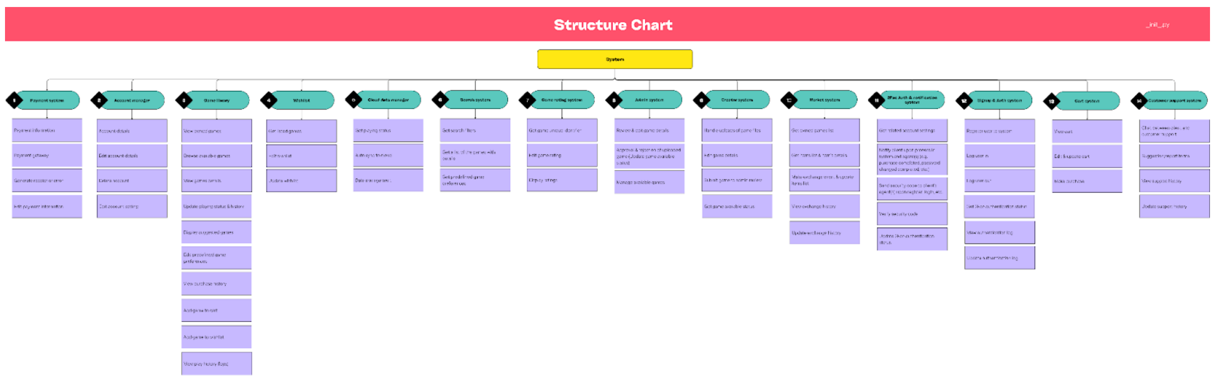
Phrase 2 Use case



Use case revision phrase 3



Functional decomposition diagram (Structure chart)



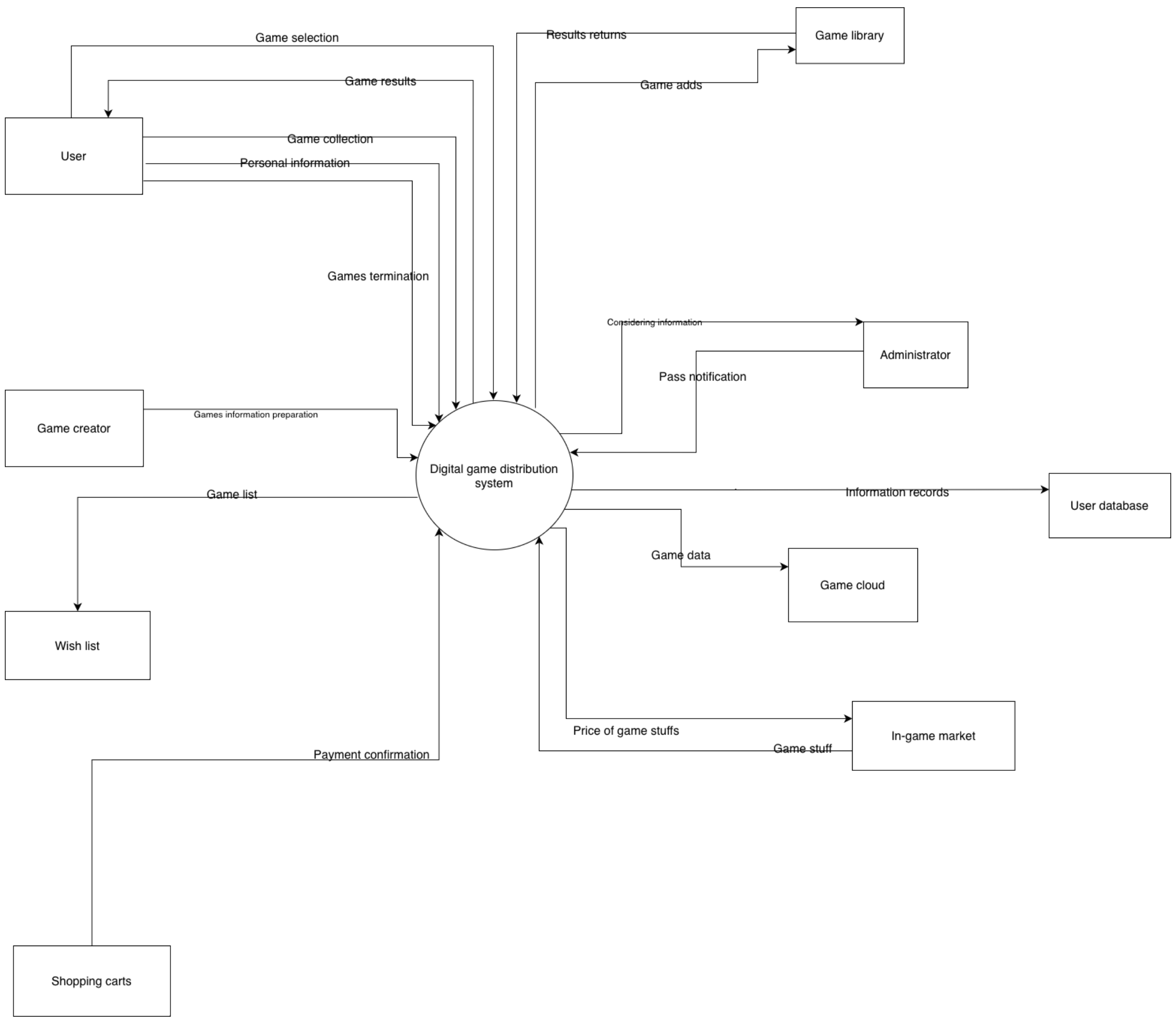
Structure explanation

<u>System</u>

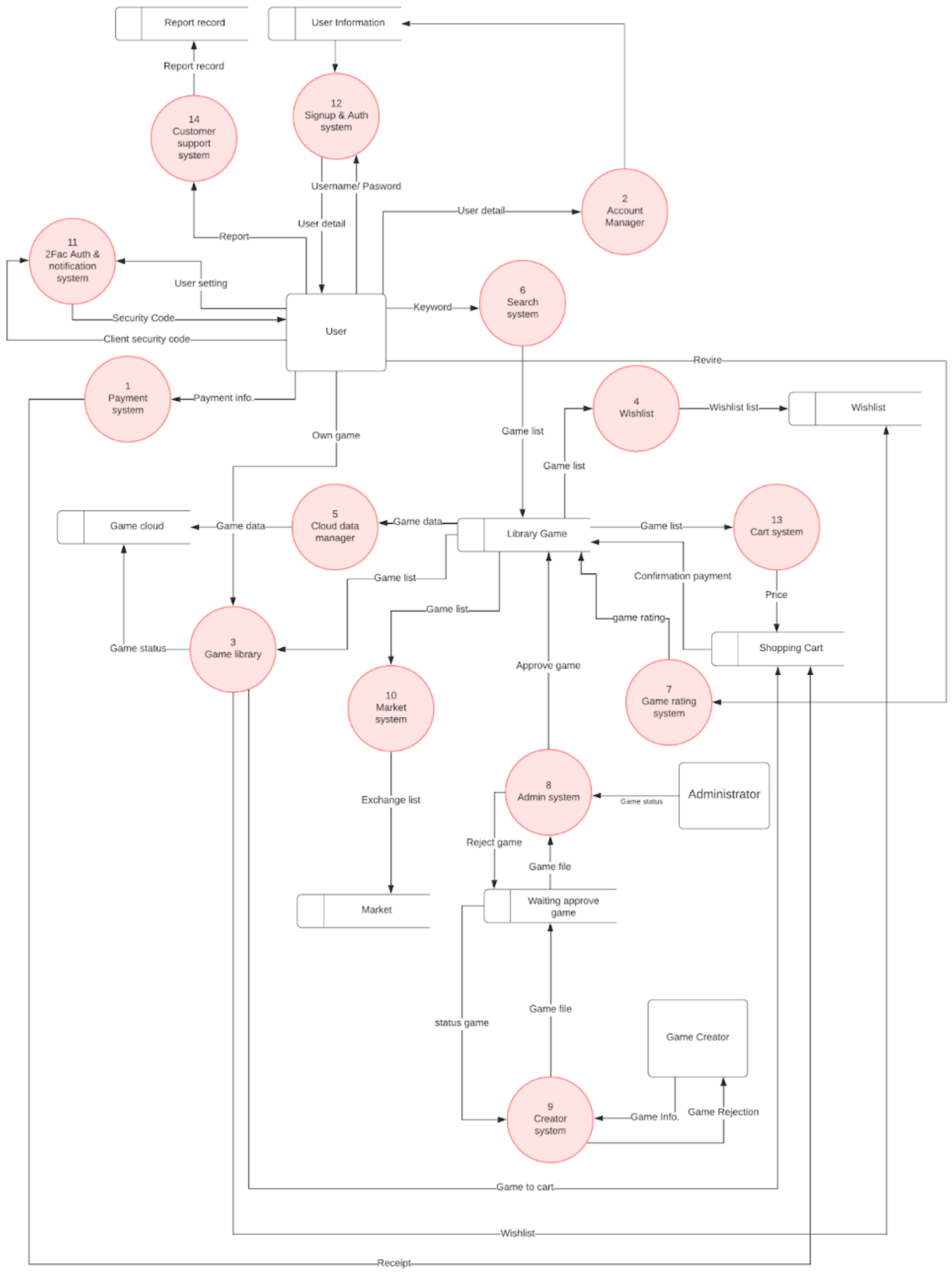
1. Payment system
    - 1.1. Payment information
    - 1.2. Payment gateway
    - 1.3. Generate receipt or error
    - 1.4. Edit payment information
2. Account Manager
    - 2.1. Account details
    - 2.2. Edit account details
    - 2.3. Delete account
    - 2.4. Edit account setting
3. Game library
    - 3.1. View owned games
    - 3.2. Browse available games
    - 3.3. View games details
    - 3.4. Update playing status & history
    - 3.5. Display suggested games
    - 3.6. Edit predefined game preferences
    - 3.7. View purchase history
    - 3.8. Add game to cart
    - 3.9. Add game to wishlist
    - 3.10. View play history (log)
4. Wishlist
    - 4.1. Get listed games
    - 4.2. Edit wishlist
    - 4.3. Update wishlist
5. Cloud data manager
    - 5.1. Get playing status
    - 5.2. Auto sync to cloud
    - 5.3. Data management

6. Search system
    6.1. Get search filters
    6.2. Get a list of the games with details
    6.3. Get predefined game preferences
7. Game rating system
    7.1. Get game unique identifier
    7.2. Edit game rating
    7.3. Display ratings
8. Admin system
    8.1. Review & edit game details
    8.2. Approval & rejection of uploaded game (Update game available status)
    8.3. Manage available games
9. Creator system
    9.1. Handle uploads of game files
    9.2. Edit game details
    9.3. Submit game to admin review
    9.4. Get game available status
10. Market system
    10.1. Get owned games list
    10.2. Get items list & item’s details
    10.3. Make exchange event & update item list
    10.4. View exchange history
    10.5. Update exchange history
11. 2Fac Auth & notification system
    11.1. Get related account settings
    11.2. Notify client upon process in system and agent(s) (e.g. purchase completed, password changed completed, etc.)
    11.3. Send security code to client’s agent(s) upon registration, login, etc.
    11.4. Verify security code
    11.5. Update 2Fac authentication status
12. Signup & Auth system
    12.1. Register user to system
    12.2. Log user in
    12.3. Log user out
    12.4. Get 2Fac authentication status
    12.5. View authentication log
    12.6. Update authentication log
13. Cart system
    13.1. View cart
    13.2. Edit & update cart
    13.3. Make purchase
14. Customer support system
    14.1. Chat between client and customer support
    14.2. Suggestion/report forms
    14.3. View support history
    14.4. Update support history

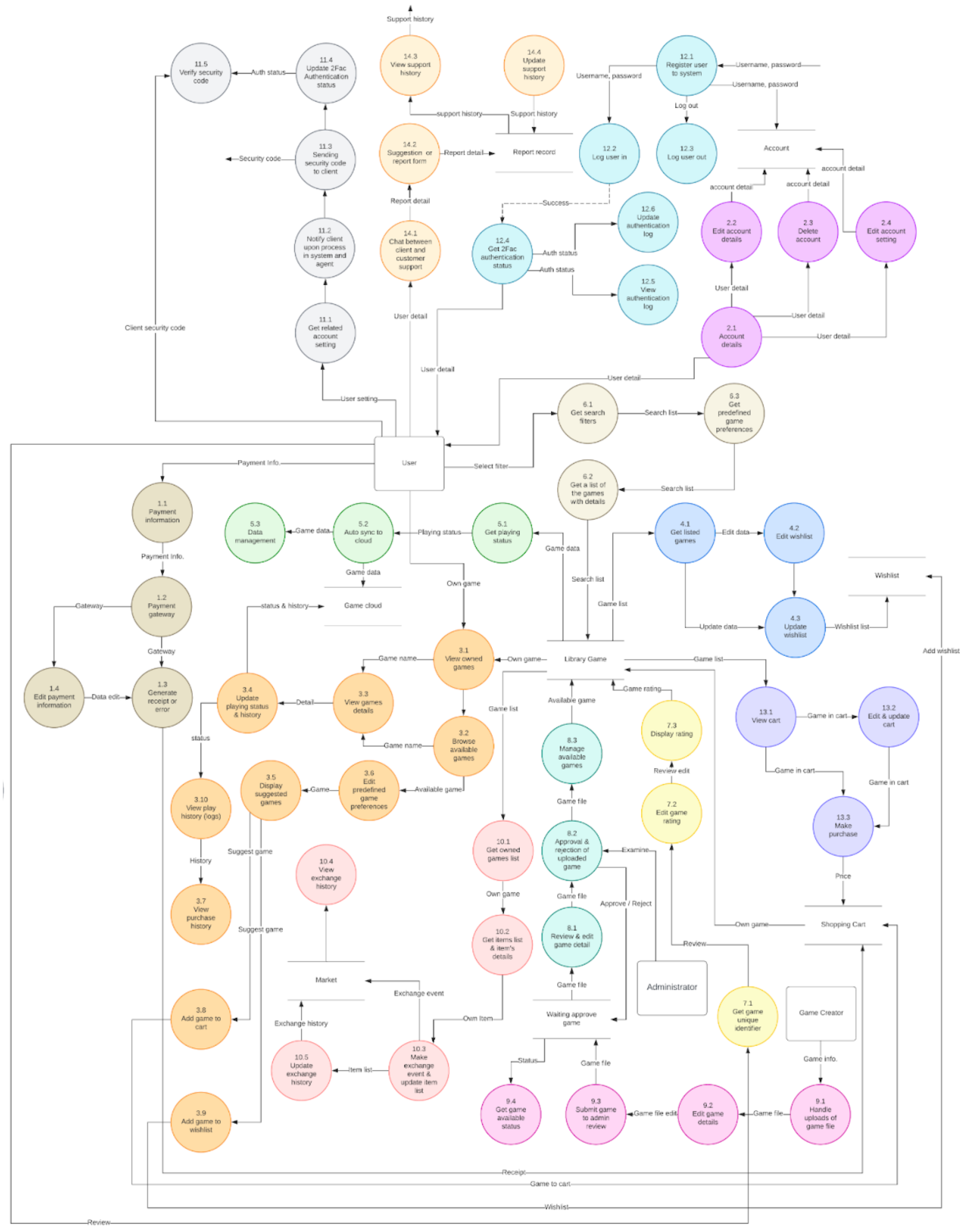
Data Flow Diagram Level 0 (Context diagram)



Data Flow Diagram Level 1



Data Flow Diagram Level 2



Prototype

We have used a Web application to implement and we are using Flask in this project, We selected process 2, 6, and some part of 12 to implement the prototype where the prototype's functionalities can be broken down into the following functionalities:

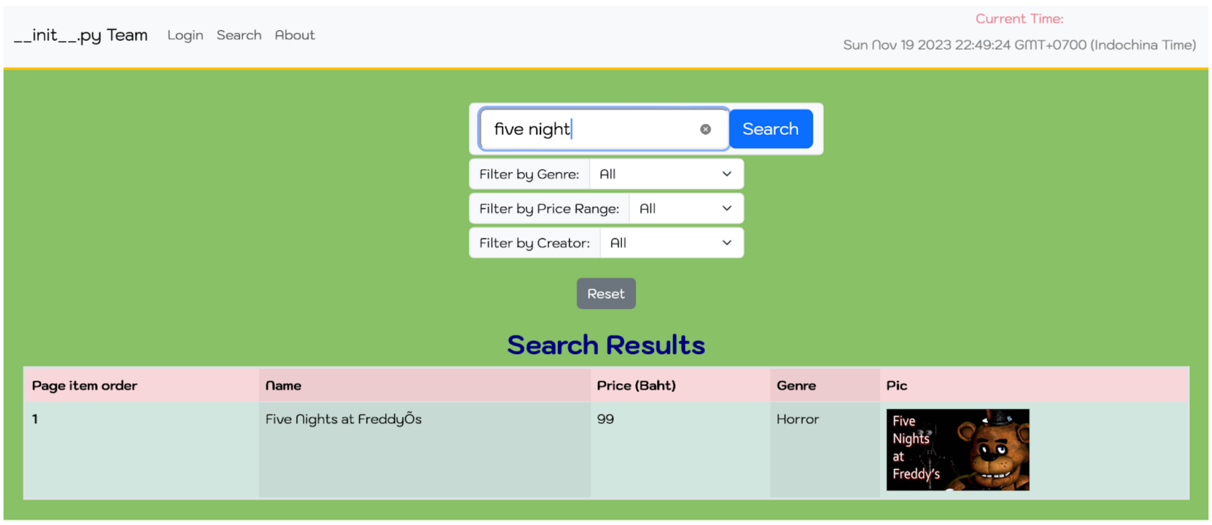
1. Account Manager
   1.1. Account details
   1.2. Edit account details
   1.3. Delete account
   1.4. Edit account setting
2. Search system
   2.1. Get search filters
   2.2. Get a list of the games with details
3. Signup & Auth system (Identity)
   3.1. Register user to system
   3.2. Log user in
   3.3. Log user out

Test case

Test ID: 1
Test name: TestValidSearch
Steps:

1. Go to http://127.0.0.1:5500/search/
2. Fill in the keyword or choose by filter that we provide
3. Click Search

Expected result: List of games that we searched for.
Actual result: List of games that we searched for.



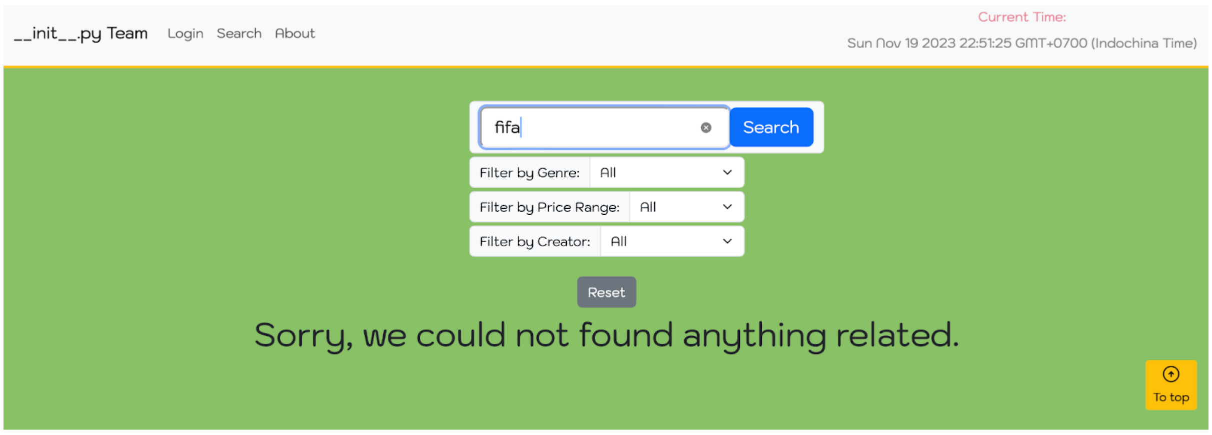
Search Results

Page item order	Name	Price (Baht)	Genre	Pic
1	Five Nights at FreddyÕs	99	Horror	Five Nights at Freddy's

Test ID: 2
Test name: TestInvalidSearch
Steps:

1. Go to http://127.0.0.1:5500/search/
2. Fill in the keyword or choose by filter that we provide
3. Click Search

Expected result: Search not found
Actual result: Search not found because that game isn't listed on our database.

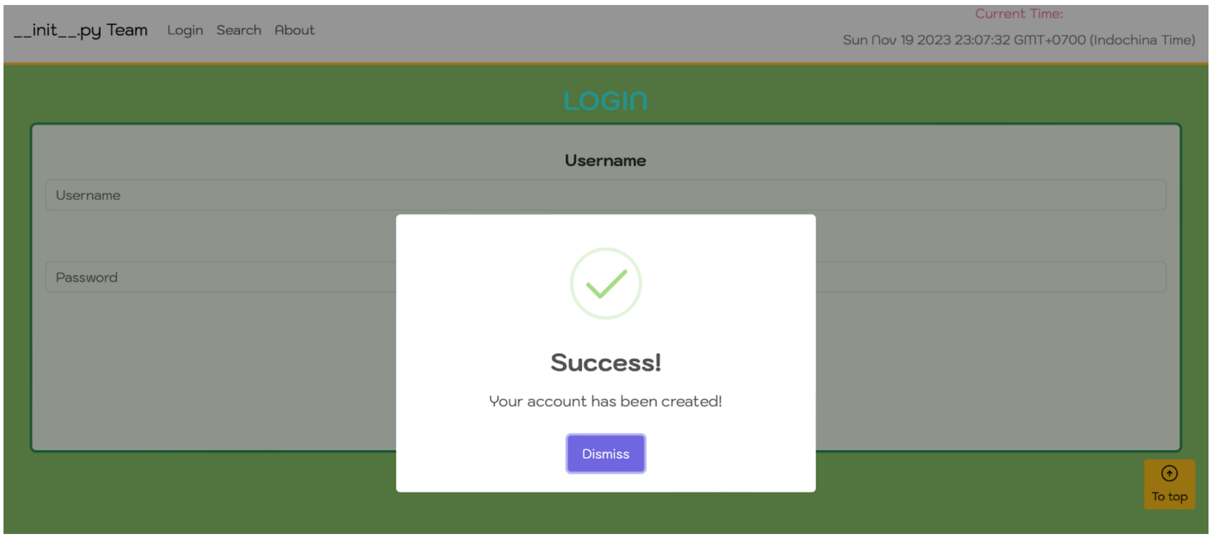


Sorry, we could not found anything related.

Test ID: 3
Test name: TestValidSignup
Steps:

1. Go to http://127.0.0.1:5500/inden-operation/signup
2. Fill in username and password that you want
3. Click create account

Expected result: success
Actual result: Success because username doesn't conflict within database

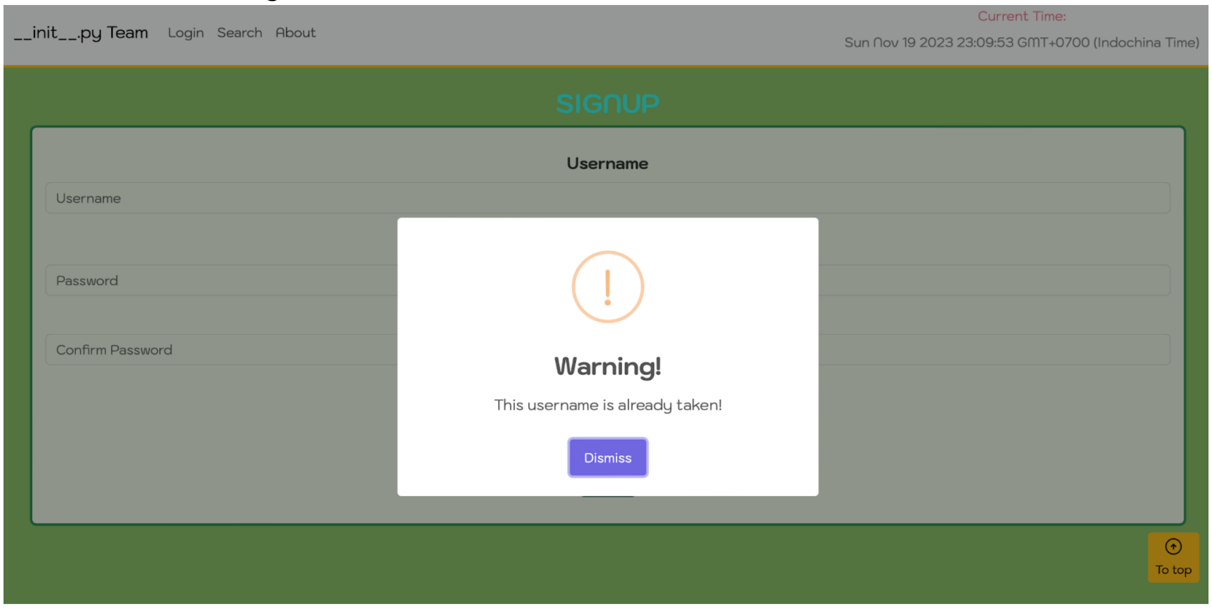


Success!
Your account has been created!

Test ID: 4
Test name: TestInValidSignup
Steps:

1. Go to http://127.0.0.1:5500/inden-operation/signup
2. Fill in username and password that you want, and try to use same username in database on purpose
3. Click create account

Expected result: Warning!
Actual result: Warning! Because username conflict within database

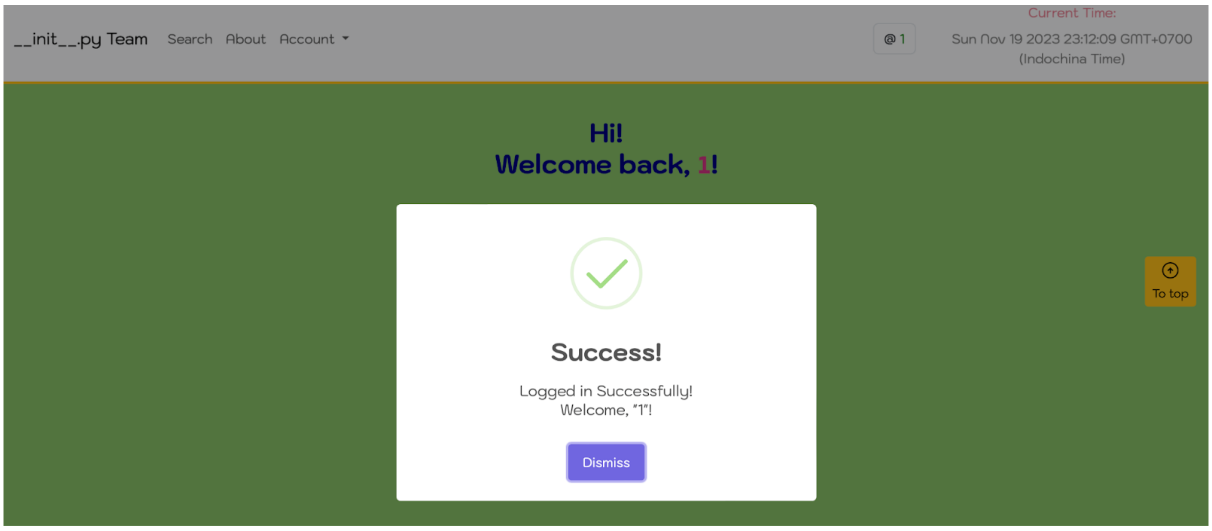


Warning!
This username is already taken!

Test ID: 5
Test name: TestValidLogin
Steps:

1. Go to http://127.0.0.1:5500/iden-operation/login
2. Fill in correct username and password
3. Click login

Expected result: success
Actual result: Success because name and password is correct compare

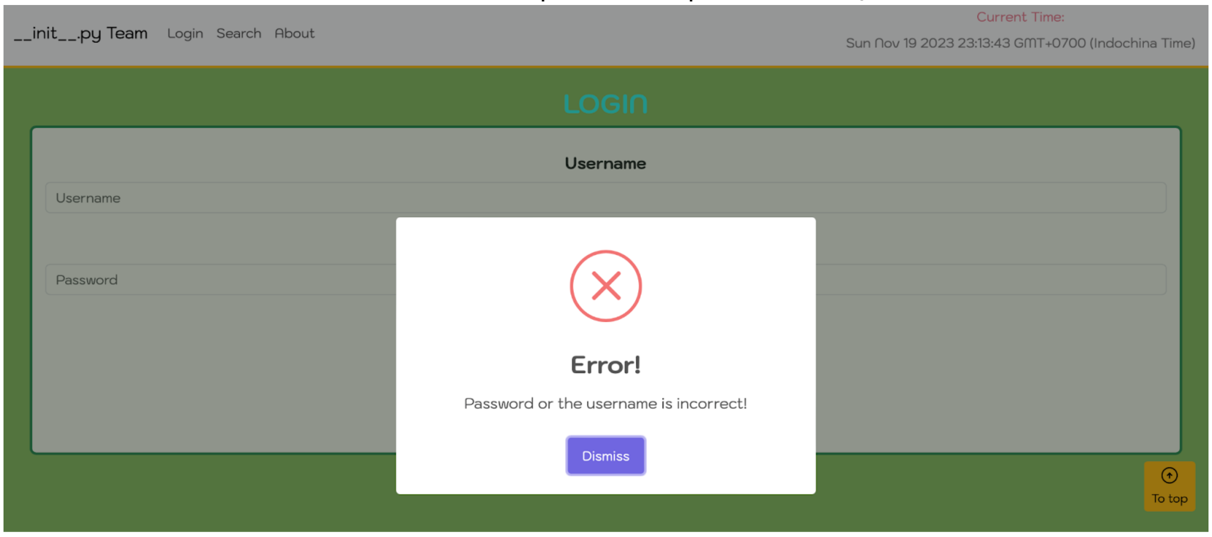


Hi!
Welcome back, 1!
Success!
Logged in Successfully!
Welcome, "1"!

Test ID: 6
Test name: TestInValidLogin
Steps:

1. Go to http://127.0.0.1:5500/iden-operation/login
2. Fill in incorrect username or password
3. Click login

Expected result: Error!
Actual result: Error! Because username or password input incorrectly



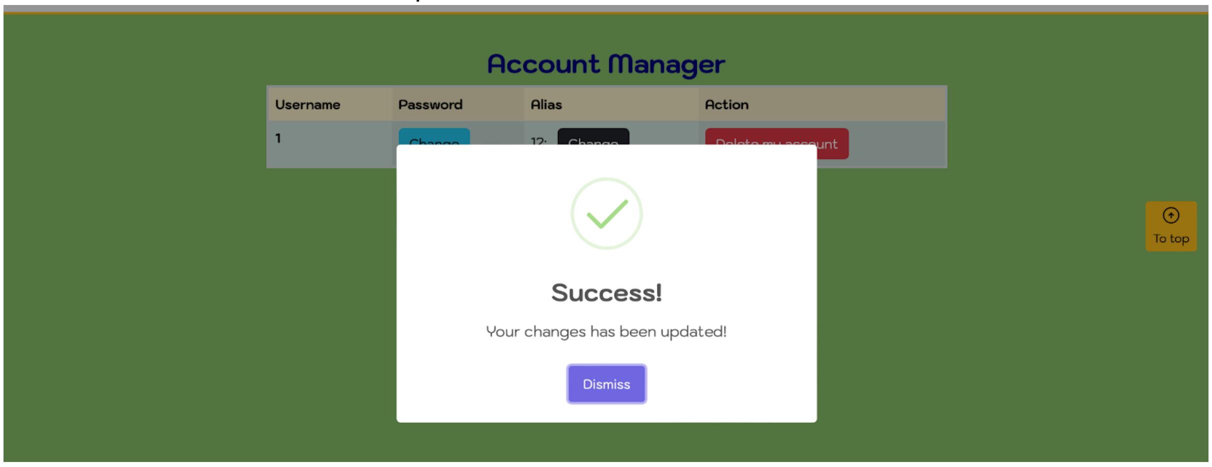
Error!

Password or the username is incorrect!

Test ID: 7
Test name: Test Alias changing
Steps:

1. Login to UI
2. Click Account
3. Click AccountManagement
4. Click change under Alias
5. Input current Alias
6. Input new Alias
7. Click submit change

Expected result: success
Actual result: success because input current Alias doesn't match current Alias

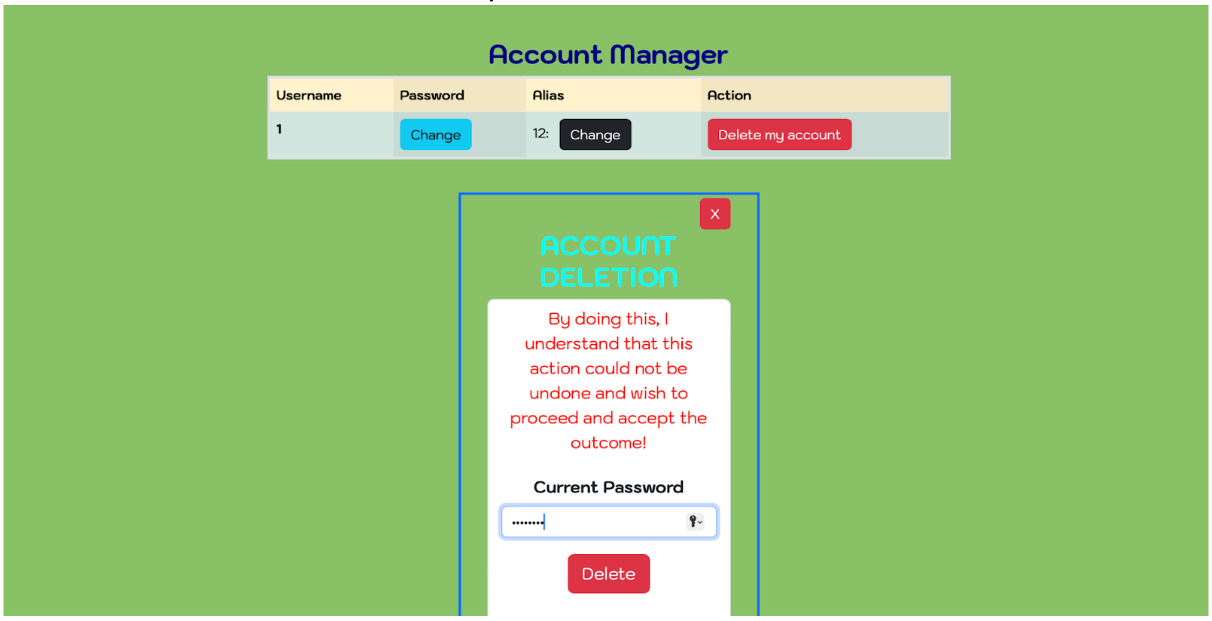


Success!

Your changes has been updated!

Test ID: 8
Test name: Test delete account
Steps:
1. Login in UI
2. Click Account
3. Click AccountManagement
4. Click Delete my account under Action
5. Enter current password
6. Click delete

Expected result: Delete successfully
Actual result: Delete success because password match, and it could be deleted



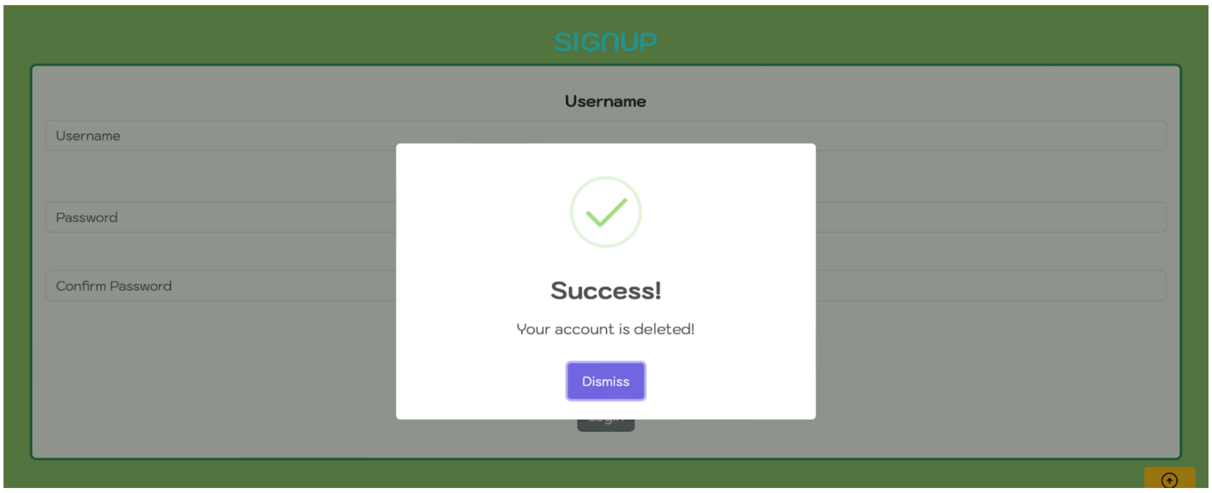
Account Manager

Username	Password	Alias	Action
1	Change	12: Change	Delete my account

ACCOUNT DELETION

By doing this, I understand that this action could not be undone and wish to proceed and accept the outcome!

Current Password



SIGNUP

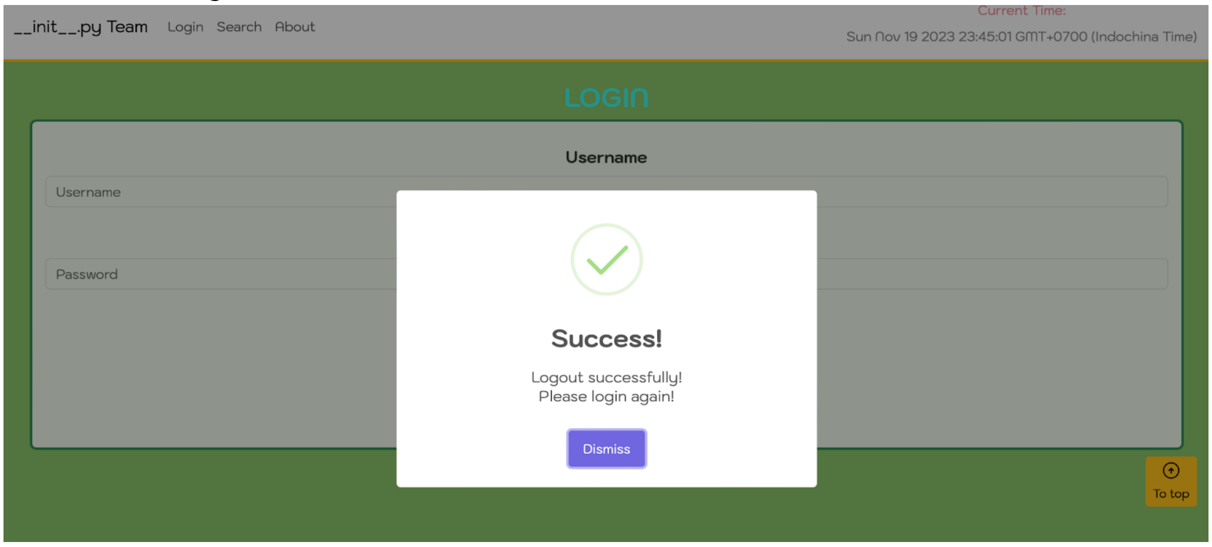
Username

Success!

Your account is deleted!

Test ID: 9
Test name: Test ValidLogout
Steps:
1. Login in UI
2. Click Account
3. Click Logout

Expected result: logout success
Actual result: logout success



Success!

Logout successfully!
Please login again!