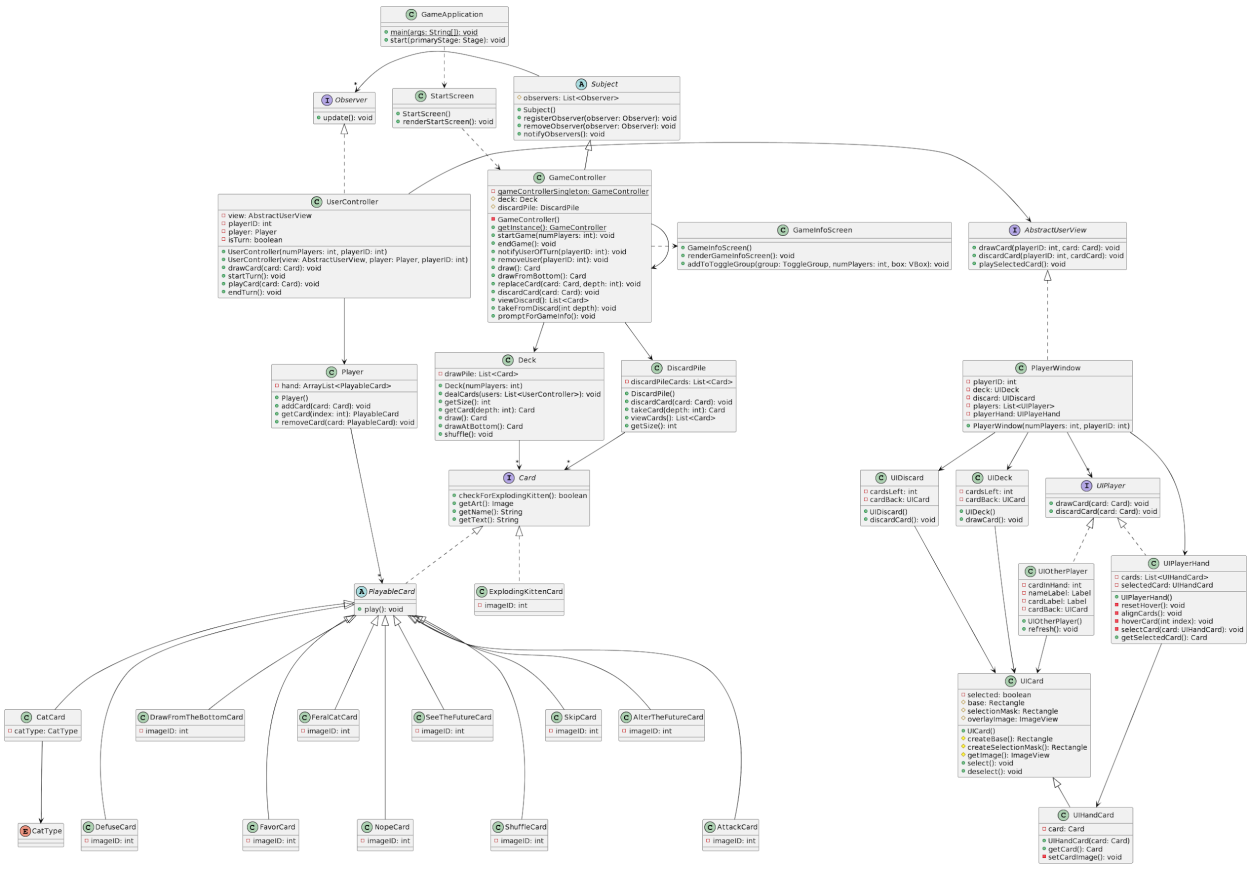
Requirements (Game Rules):

Our software will simulate the majority of elements of gameplay from the popular card game, Exploding Kittens.
The rules can be found here: https://www.explodingkittens.com/pages/rules-kittens-party
Features:

- Starting screen
- Multiple screens (1 for each player)
- Ability to play with up to 10 players
- Compatible with three languages:
    - English
    - Japanese
    - Klingon
- Tracks:
    - The deck
    - The discard pile
    - Players' hands
    - A player's selected cards
- Ability to:
    - Draw cards
    - Play cards with varying effects
        - Attack
            - End turn, force next player to take two turns
        - Skip
            - End turn immediately
        - Cat
            - Play as a pair to steal a card from a player of their choosing
        - Defuse
            - Avoid losing the game by defusing an Exploding Kitten
        - Nope
            - Cancel another player's special card
        - See the Future
            - Peek at the next three cards in the draw pile
        - Shuffle
            - Shuffle the deck
        - Favor
            - Choose a random card from an enemy's hand
        - Draw from Bottom
            - Draw a card from the bottom of the deck
    - Eliminate a player by drawing an Exploding Kitten card
    - Determine a winning player to be the last one standing

Architecture/Design:

- Refer to UML Diagram below:



[Link to PlantUML here]

Planned Final Functionality

Milestone 2

- Dynamic window sizing (Braxton, Dalton)
    - Dynamic windows scaling was initially pursued as a larger feature to be completed by Dalton and Braxton, however the means by which that could.should be accomplished were unpredictable, and this seemed to be a difficult problem to solve. We were unable to get any predictable implementation working, and we decided that this feature may need a temporary or permanent rain check.
- Sort cards within the hand

    - Cards were shown in sorted alphabetical order in each player's hands. Previously, cards were kept in the hand in the order that they were drawn, which made it difficult to tell when you had duplicates, which is a useful fact to know during gameplay. When sorting, duplicates are placed adjacent to each other in the hand, and it is easier for a user to find the card they are looking for
- Display of the various functions of decks displayed on the UI (Andrew)
    - Add labels to the various UI components to better indicate their purpose and differentiate themselves. For example, although the discard pile, deck, and elements for other players all have the same image, they now have different labels to help set them apart.
- Display of the current turn state for each player (Blanco)
    - Each user's window should display to the user information regarding their turn state. Specifically, the window should inform the user that they are still waiting for their turn (if it is not currently their turn) or that they are able to perform an action now (if it is currently a turn).

Milestone 3

- Ability to upload custom art for cards (Dalton)
    - The user will be able to upload an image to the system and specify the type of card that they'd like their image to be associated with.
- "Pass and play" mode (Andrew)
    - The dominant view on the screen automatically changes to the view associated with the player whose current turn it is, as well as during opportunities where a player may perform a counteraction (e.g. player is able to "Nope" an action).
- Key bindings (Braxton)
    - Binding card selection to the keyboard numbers, and playing cards with the space bar or enter key
- Spectator mode (Nathaniel)
    - The system generates a view of the game that displays more information than available to the participants, including the next few cards on top of the Draw pile the hands of every player.

Milestone 4 (All team members on all tasks)

- Finalize all software artifacts for design, requirements, and architecture
- Complete remaining work on test suite
- FInalize code for last build