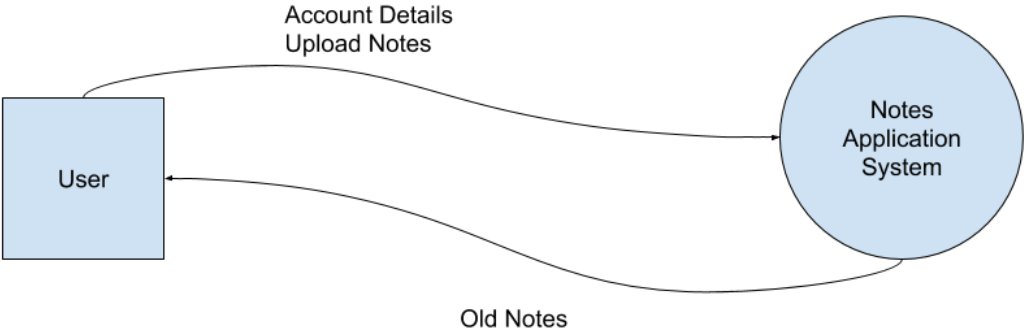
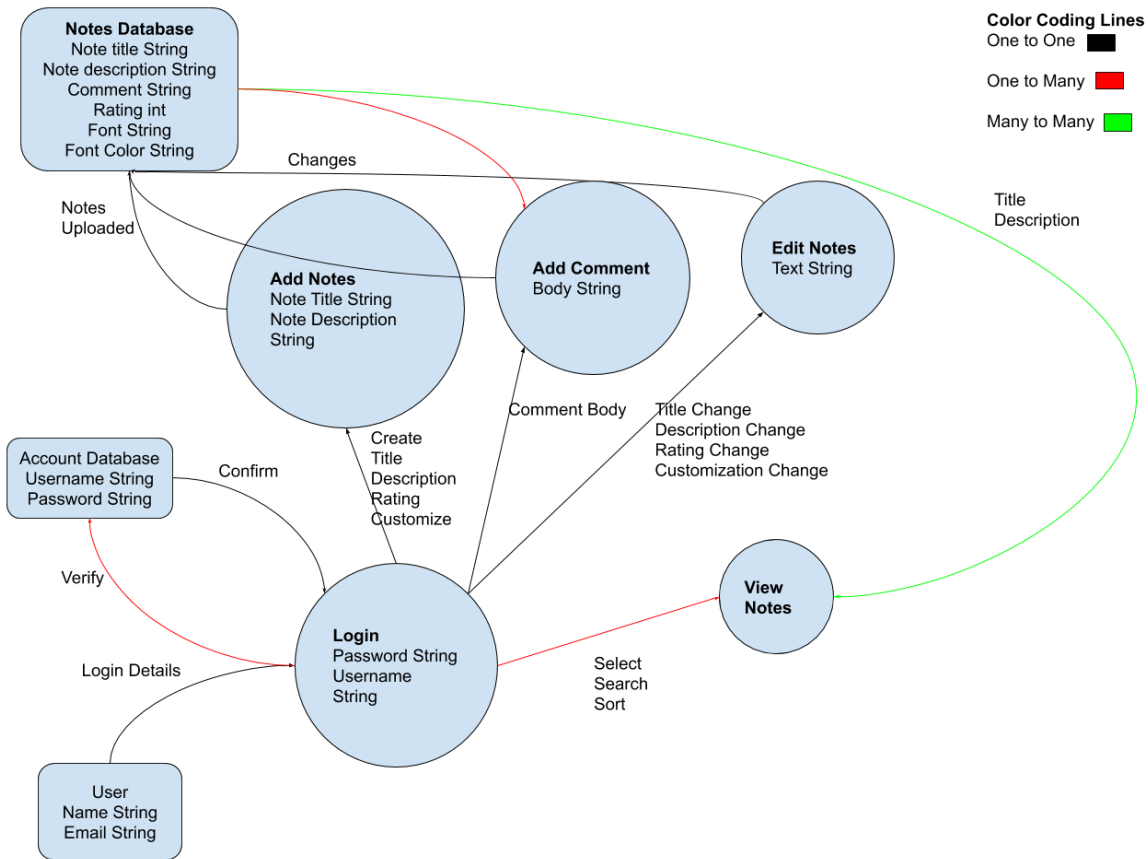


Complete DFD and model relationship diagrams





Models Pseudocode

```
class user(Parent):
    __tablename__ = 'user'
    id = Column(Integer, primary_key=True)
    note = Column(String)
    rating = Column(String)
    comment = Column(String)
    children = relationship("notes", back_populates="parent")
```

```
class notes(Parent):
    __tablename__ = 'notes'
    id = Column(Integer, primary_key=True)
    note = Column(String)
    rating = Column(String)
    comment = Column(String)
```

```
children = relationship("note", back_populates="parent")
```

```
class note(Child):
    __tablename__ = 'note'
    id = Column(Integer, primary_key=True)
    title = (String)
    body = (String)
    font= Column(String)
    fontcolor = Column(String)
    fontsize = Column(Integer)
    comment = Column(String)
    rating = (Integer?)
    parent_id = Column(Integer, ForeignKey('notes.id'))
    parent = relationship("notes", back_populates="children")
    children = relationship("rating, comment", back_populates="children")
```

```
class rating(Child):
    __tablename__ = 'Rating'
    id = Column(Integer, primary_key=True)
    rating = (Integer)
    parent_id = Column(Integer, ForeignKey('parent.id'))
    parent = relationship("note", back_populates="rating")
```

```
class comment(Base):
    __tablename__ = 'comment'
    id = Column(Integer, primary_key=True)
    body(String)
    parent_id = Column(Integer, ForeignKey('parent.id'))
    parents = relationship("note", back_populates="comment")
```

All user stories (e.g., Screenshots of Trello cards)

Feature 1 - List notes

in list [Done](#)

Description [Edit](#)

As a user, I want to be able to see the list of notes that I have so that I can use other features if I need it.

○

Feature 2 - View a note

in list [Done](#)

 Description [Edit](#)

As a user, I want to be able to view a note so that I see the contents of the note.

Feature 3 - Create a note

in list [Done](#)

 Description [Edit](#)

As a user, I want to be able to navigate the website so that I can create a new note.

Feature 4 - Edit a note

in list [Done](#)

 Description [Edit](#)

As a user, I would like to be able to go back and change, add, or delete what is written in a note.

Feature 5 - Comment on a note

in list [Done](#)

 Description [Edit](#)

As a user, I would like to comment on a note to add additional information and clarify anything that needs to be done.

Feature 6 - Delete a note

in list [Done](#)

Description [Edit](#)

As a user, I would like to be able to discard notes that I don't want or need anymore.

Feature 7 - Register a user

in list [Done](#)

Description [Edit](#)

As a user, I would like to create an account that will store all my notes for easy access.

Feature 8 - User sign-in

in list [Done](#)

Description [Edit](#)


As a user, I want to be able to sign into my account that has all my previous notes saved.

Feature 9 - Delete comment


in list [Done](#)


Description [Edit](#)

As a user, I would like to be able to delete any comments that I made to clear the clutter on my notes.


-  **Feature 10 - Searching**

in list [Done](#)

 **Description** Edit

As a user I would like to be able to search for a specific note in order to reduce time spent scrolling and manually searching for the note.
-  **Feature 11 - Customizing styles based on user preferences (e.g., user select theme (color, image, font..))**
✕


in list [Done](#)


 **Description** Edit

As a user, I want to switch between different themes, a light and a dark theme.


SUGGESTED

⚙️

 Join

ADD TO CARD
-  **Feature 12 - Rating (5-star or thumbs up/down)**

in list [Done](#)

 **Description** Edit

As a user, I want to be able to rate a note.

Description of current features and any features not yet implemented.

- All the features that are implemented are all the base features.
 - List Notes
 - Allows the user see the a list of all the notes they created.
This feature was implemented.
 - View a Note

- Allows the user to see a note of theirs that they had created. This feature was implemented.
- Create a Note
 - Allows a user to create a new note which will also be added to the list of their current notes. This feature was implemented.
- Edit a Note
 - Allows the user to modify one of their notes. This feature was implemented.
- Comment on a Note
 - Allows the user to add a comment onto a note to clarify if anything else needs to be done. This feature was implemented.
- Delete a Note
 - Allows the user to delete a note of theirs which gets removed from their list of notes. This feature was implemented.
- Register a User
 - Allows a user to create an account on our application. This feature was implemented.
- User Sign-in
 - Allows the user to log into their account to have access to their notes. This feature was implemented.
- Delete a Comment (Additional Feature)
 - Allows the user to delete a comment from a note. This feature was implemented.
- Search Bar (Additional Feature)
 - Allows the user to find a note using a search bar. This feature was implemented.
- Dark Mode Theme (Additional Feature)
 - Allows the user to switch between a light and dark mode. This feature was implemented.
- Rating a Note (Additional Feature)
 - Allows the user to rate a note. Not fully implemented, made an attempt to implement, but does not fully work.

Feature Testing

- Description of the acceptance criteria required to test and verify the application (features and scenarios) (Gherkin/Cucumber) and any other testing frameworks you used (e.f. unit)

- We used different methods to test the application, the features and scenarios. We used unit test cases and manually tested the features in the application to make sure they matched the requirements of the scenarios.
- Manual testing included some of the following:
 - Trying to register a currently active account
 - Adding comments of different length, characters, values.
 - Searching for all likely keywords such as individual characters, punctuation, and white space.
 - Once dark mode was enabled, we navigated to every page and refreshed the pages to make sure it was consistent.
 - Deleted comments with various lengths, values and amounts.
- A user guide for all of the currently implemented features (explain to a user how to use your application)

Black Jack Notes App User Guide

Welcome to Black Jack notes app, this is a detailed yet simple guide to become a Black Jack master!

- Account Creation
 - To get started let's create an account.
 - Once you sign up you will automatically be logged in to the notes page.
- Navigation Bar
 - Home Tab – This is the home page with application information.
 - Notes Tab – View all your notes by clicking this tab.
 - Create Note – Create new notes by accessing this tab.
 - Logout – This tab will log you out.
 - Login – This tab will take you to the login page.
 - Change mode – Allows you to switch between light and dark mode, available in the Notes page.
 - Search Bar – Input keywords to locate a specific note.
 - Go button – Clicking this will load the notes with keywords inputted in the search bar.
- Login Page
 - Input your credentials.
 - Click the login button.
 - If the credentials provided are correct you will be granted access to Black Jack notes app.

- If your credentials are incorrect you will be prompted to re input them.
- Home Page
 - This page has the introductory content for the website.
 - The My Notes button will take you to the My Notes page.
- Notes Page
 - You will be viewing all your notes on this page.
 - By selecting the edit option, you will be relocated to the edit page for the note you have chosen
 - By clicking the notes name you will be relocated to the corresponding notes page.
 - The ratings feature is available on this page. If you deem a note important select the star rating for the note to make it stand out from others.
 - You will be able to do the following in this page:
 - Delete your notes by clicking the delete button.
 - Switch between Dark and Light mode by clicking change mode button
 - Locate specific notes based on the input provided in the search feature
- New Note Page
 - There are two inputs for this page, one for the title of the note, other for the contents of the note.
 - Once you filled out the form, click the Create Note button.
 - The note will now appear in the notes page.
 - If you choose you have the option to logout from this page.
 - If you choose you have the option to go back to your notes from this page.
- Edit Note Page
 - There are two inputs for this page, one for the title of the note, other for the contents of the note.
 - Once you're satisfied with the changes, click the Update Note button.
 - The updated note will now appear in the notes page.
 - If you choose you have the option to logout from this page
 - If you choose you have the option to go back to your notes from this page
- Note Page
- You will be redirected here when you click on the notes name on the notes page.

- In this page you will be able to view the following:
 - Note Title
 - Note Text
 - Note Create Date
 - Comments related to the note
- To create a comment simply fill out the comment text area and click the create button. Your new comment will now be visible on this page.
- Each comment is related to the corresponding note that it is created in, so do not worry about comments being duplicated in other notes.
- If you chose to delete the comment simply click the delete button located in the action column in the comments table.

Ethical Considerations

- A thorough analysis of ethical considerations and how you have addressed them or plan to address them.
 - Without an ethical consideration the user could have a system that is deployed that contains bugs, privacy issues, accessibility problems, the system itself not being accurate. Some of the ethical considerations that need to be considered is that the user has their own private data, no information is shared with anyone. This something needs to be addressed during the development of the application. Implementation of an email validator will more accurately register a user, since only the fields that are entered on the register are validated to input requirements. One thing that needs to be done after deployment of the application is to make sure the user has accessibility to the application when the user needs it.

Feature Not Implemented

- Make note of any features not yet implemented, if any!
 - The only feature we were not able to fully implement is the rating feature, there is some minor bug in the code itself possibly in the linking of the Python and HTML code.