Terminal App T1-A3

FilmSpot: a terminal program allowing users to search, select, get recommendations and be tested on the top 1000 greatest movies ever made

Video walkthrough - https://vimeo.com/753217247

Logic Design

Class "App_dataframe"

Class variable: IMBD database stored as a class variable

Staticmethods (I.e. class functions)

1. Main menu function (access all staticmethods in program)
2. Database Search function (accesses and manipulate the class variable)
3. Trivia game function (accesses class variable)

Class "User"

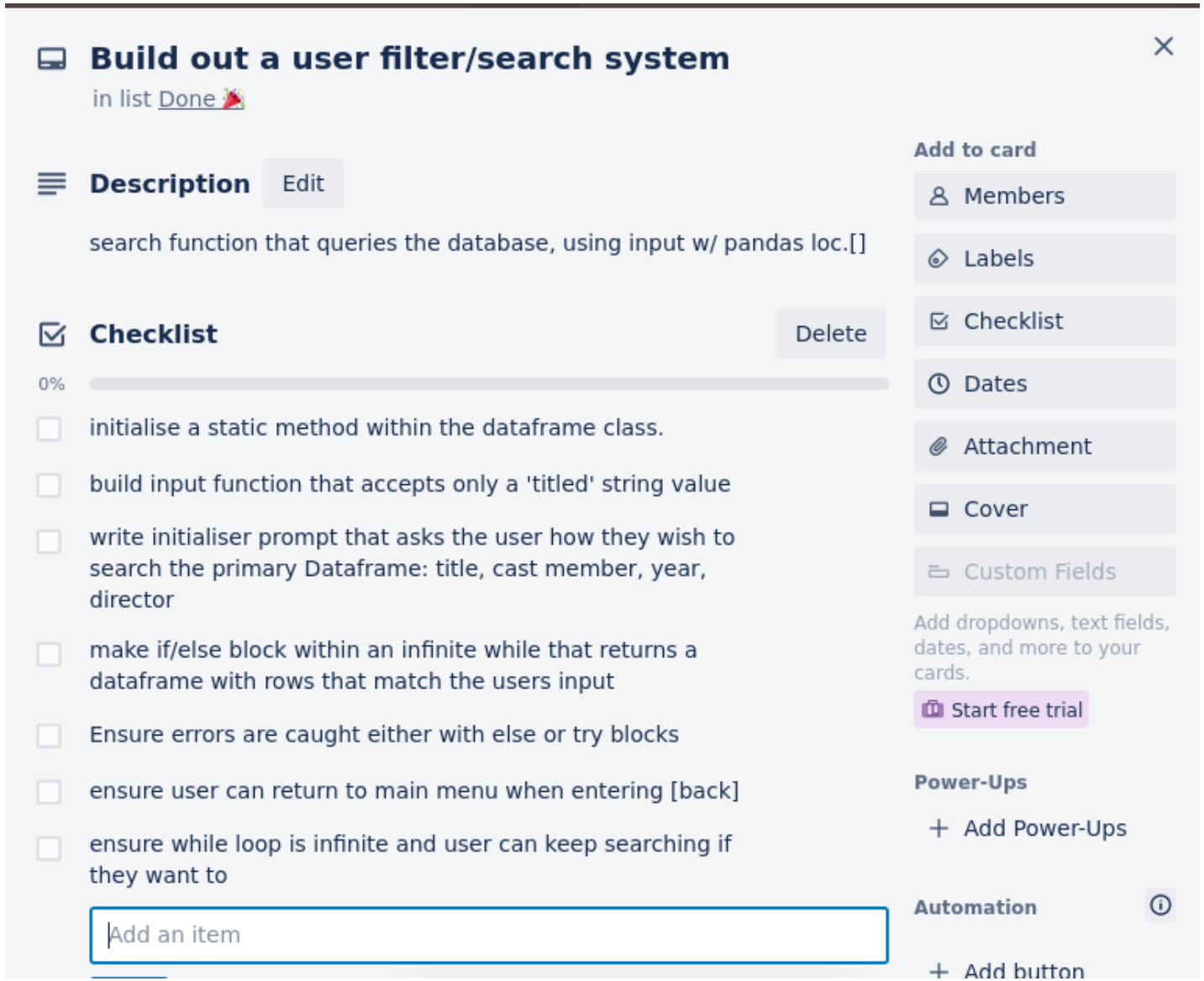
IMBD database inherited from parent

Instance variables: self.username, self.fav_movies

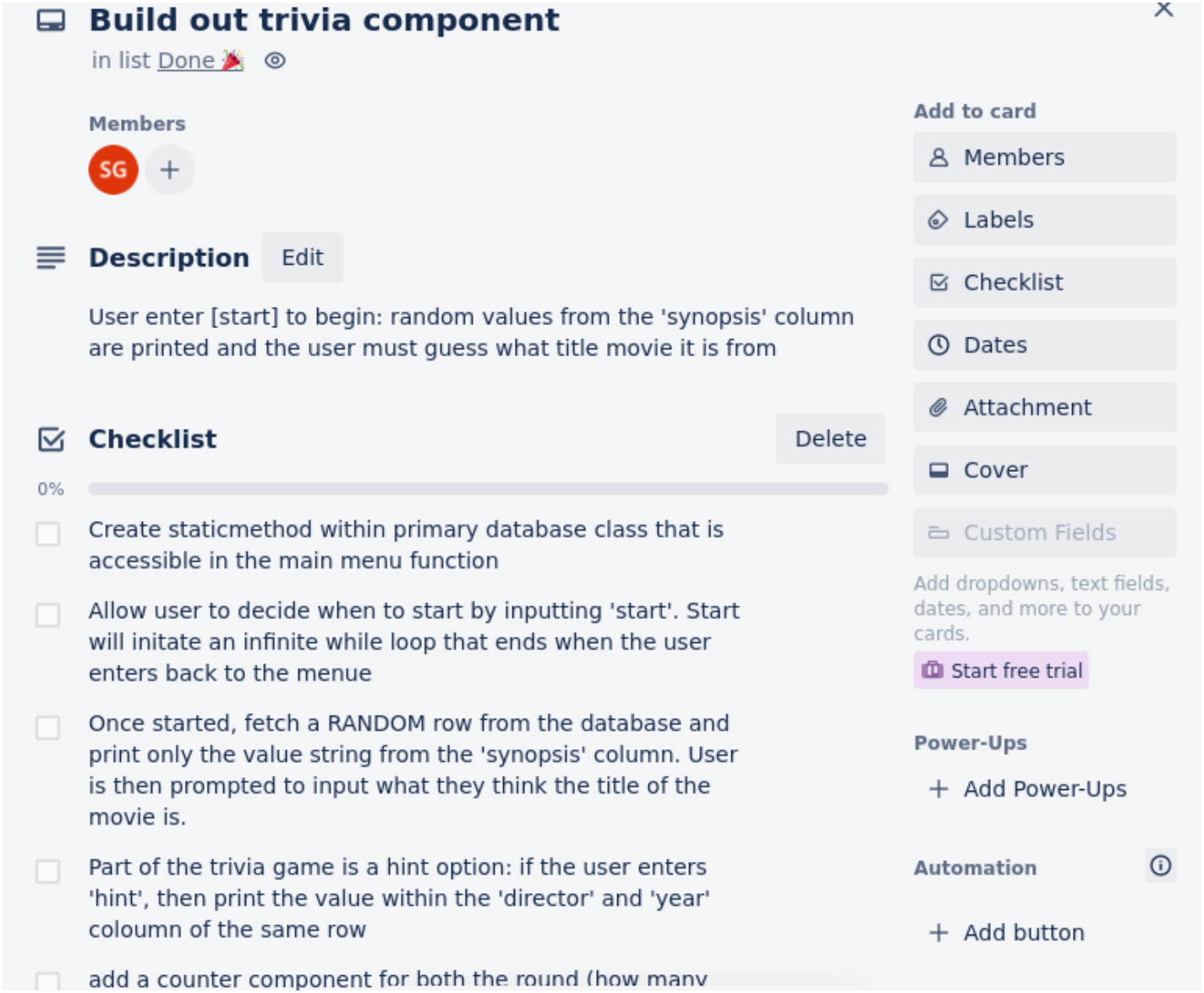
Staticmethods (I.e. sub-class functions)

1. Create/store user instance (+access all class functions)
2. Reccomendation functioin + Store reccomended movies into User instance
3. Display user watchlist

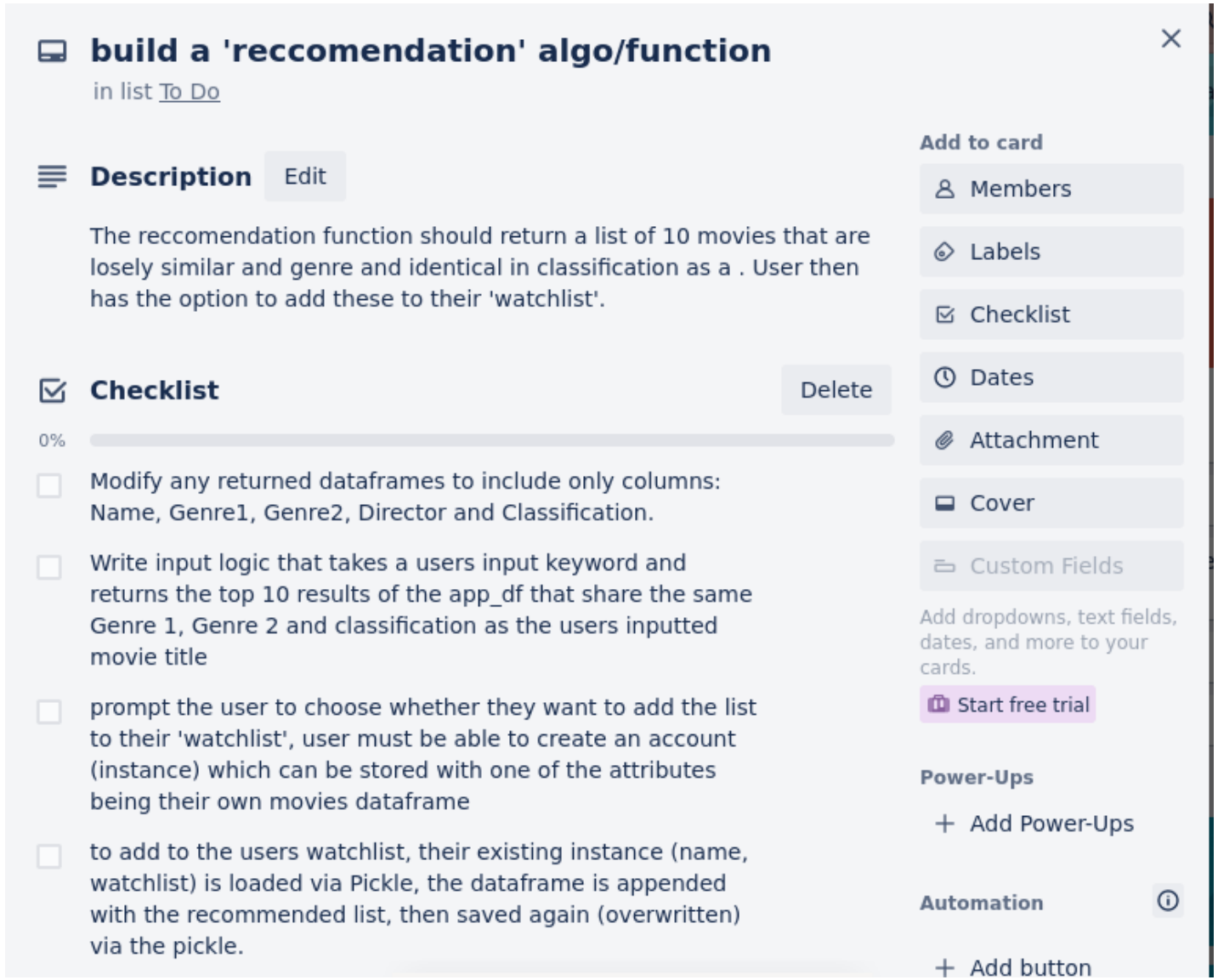
Search/filter function checklist (trello card)



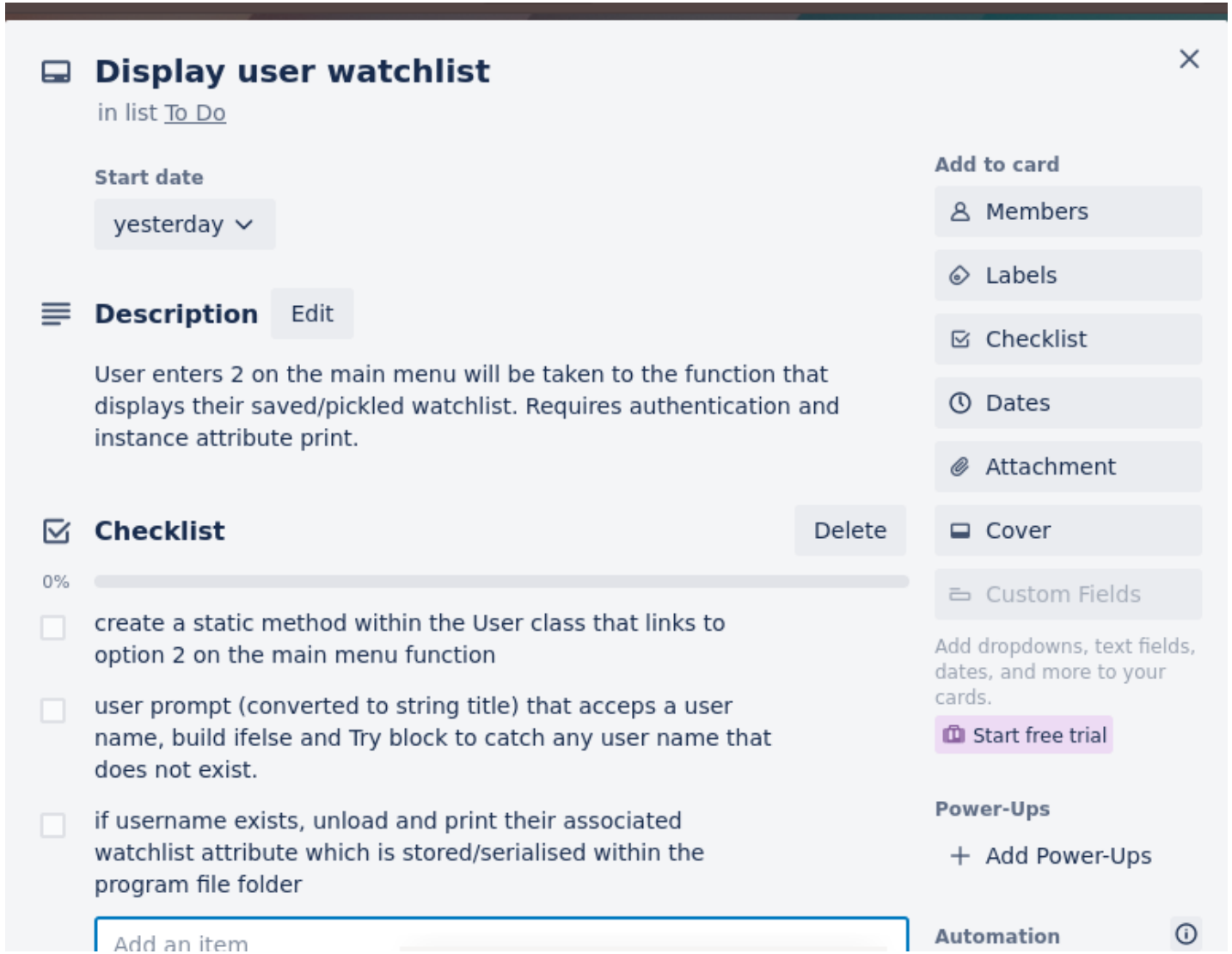
Trivia function checklist (trello card)



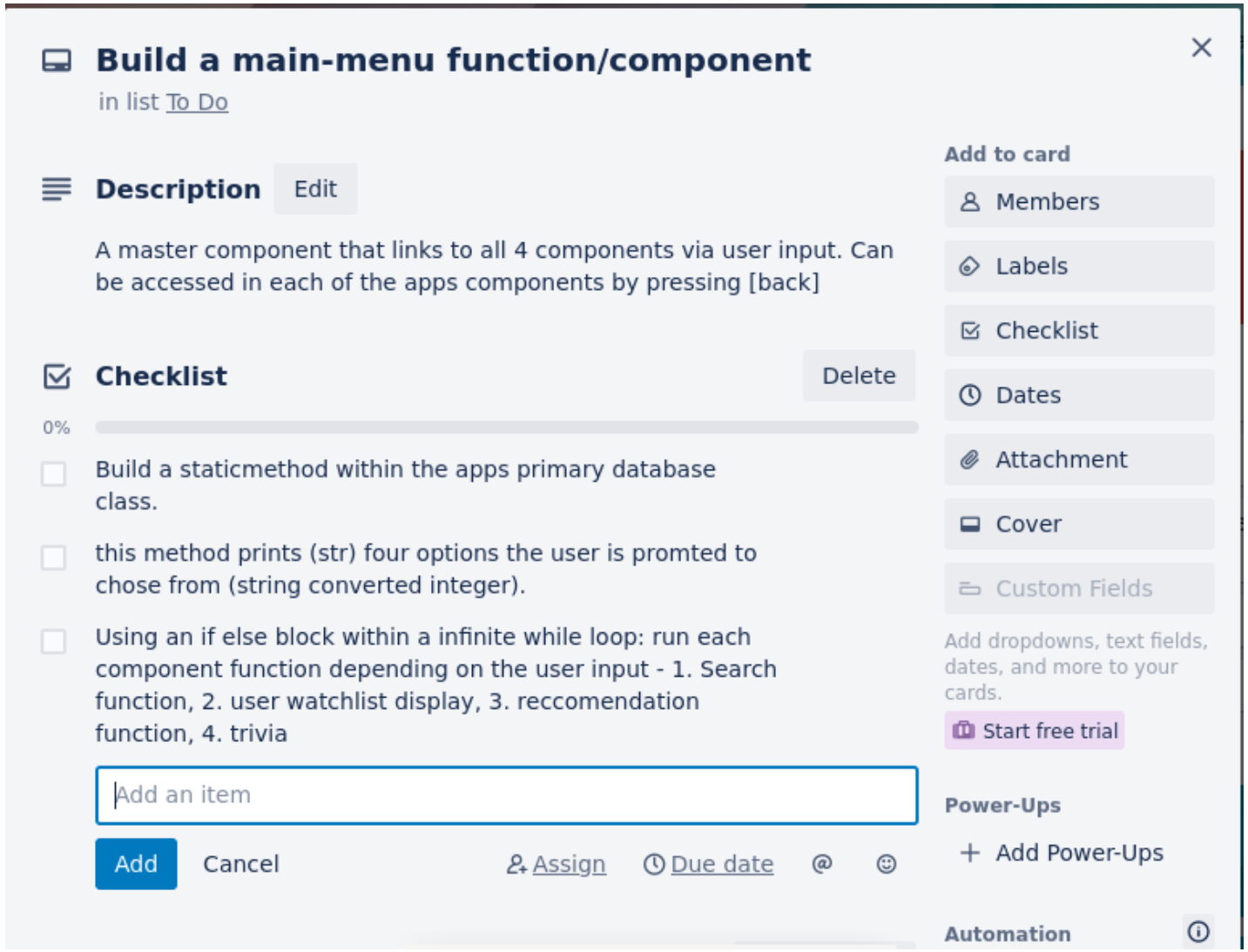
Recommendation + account creation component checklist (trello card)



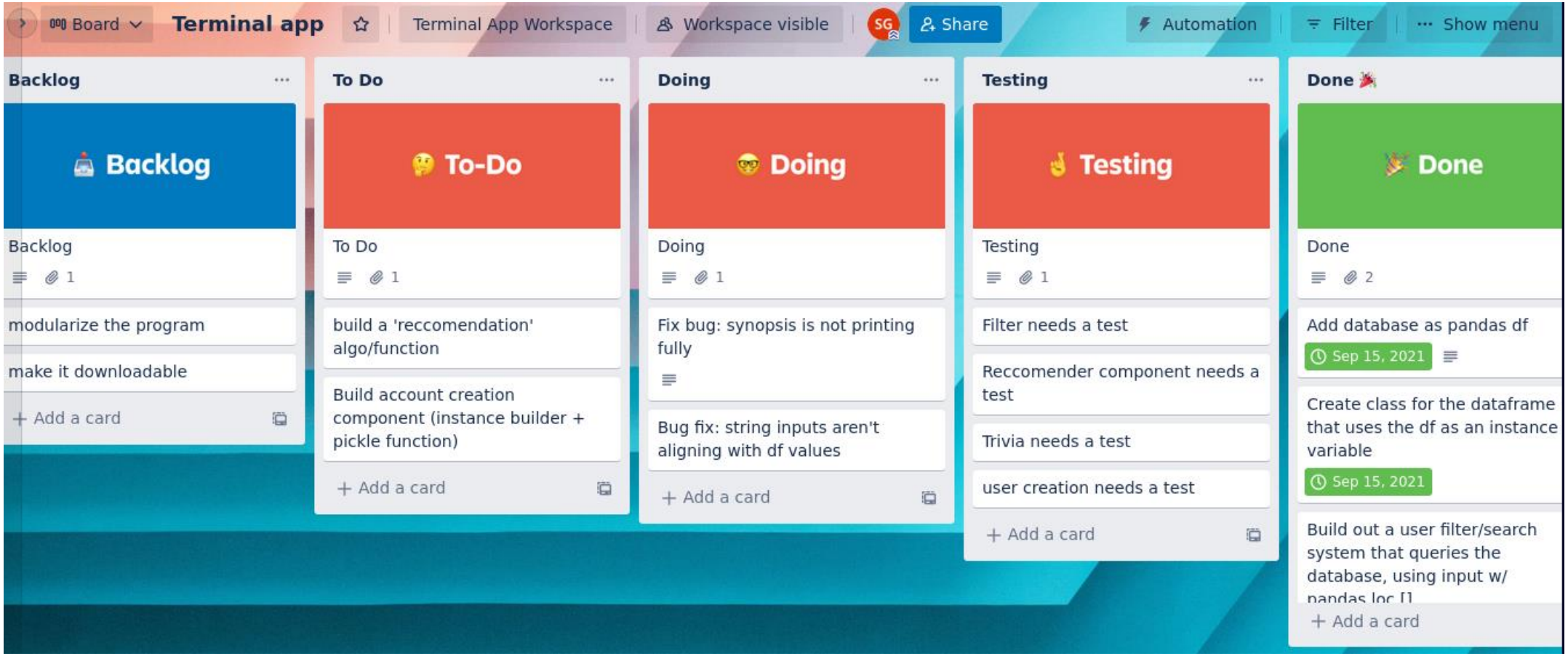
View Watchlist component checklist (trello)



Main menu checklist



Implemented via Kanban on *Trello*



(Early stage kanban board)

Final notes/caveats/challenges…

- Only have implemented 3 Try block error handlers throughout the app. This was due to how primative the user inputs (e.g. no dealing with integers) and (lack of) 'algorithms' are - i.e. exceptions weren't warranted for each conditional block.
- Fell a bit behind OOP content. Where I usually use challenges to grasp concepts I've had to use this assignment as a learning exersise:
  - This meant I had less time on refactoring and creating modular code as well as implementing more robust tests (e.g. putting my 2 classes in separate modules – it kept causing a circular import error I need to learn more about)
  - **End result** is I have learned a solid foundation on OOP and learned enough to know shortcomings in the code/what to improve for next time