Grocery List App

By Peemin Chen and
Margaret Wheeler

Link to Demo

Video:

https://github.com/peemin/csci5448/blob/master/GroceryListApp_Video.swf

Link to website:

https://grocerylist.herokuapp.com/

What we did

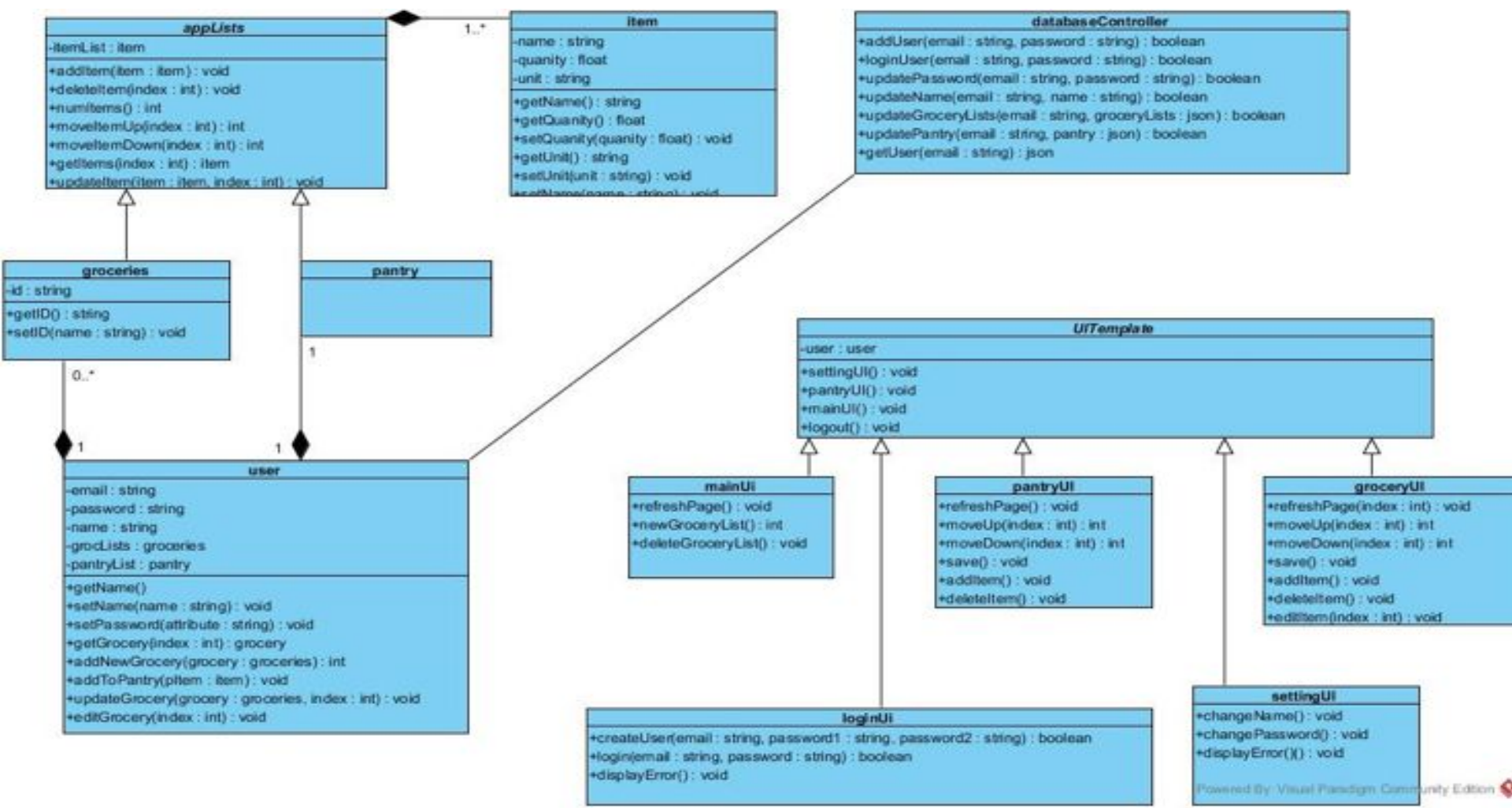
- Created a grocery list website app that lets users create digital shopping lists
  - Lists are attached to user by their sign in credentials
    - User can create multiple lists
    - Delete lists
    - Each list can contain 1 or many items

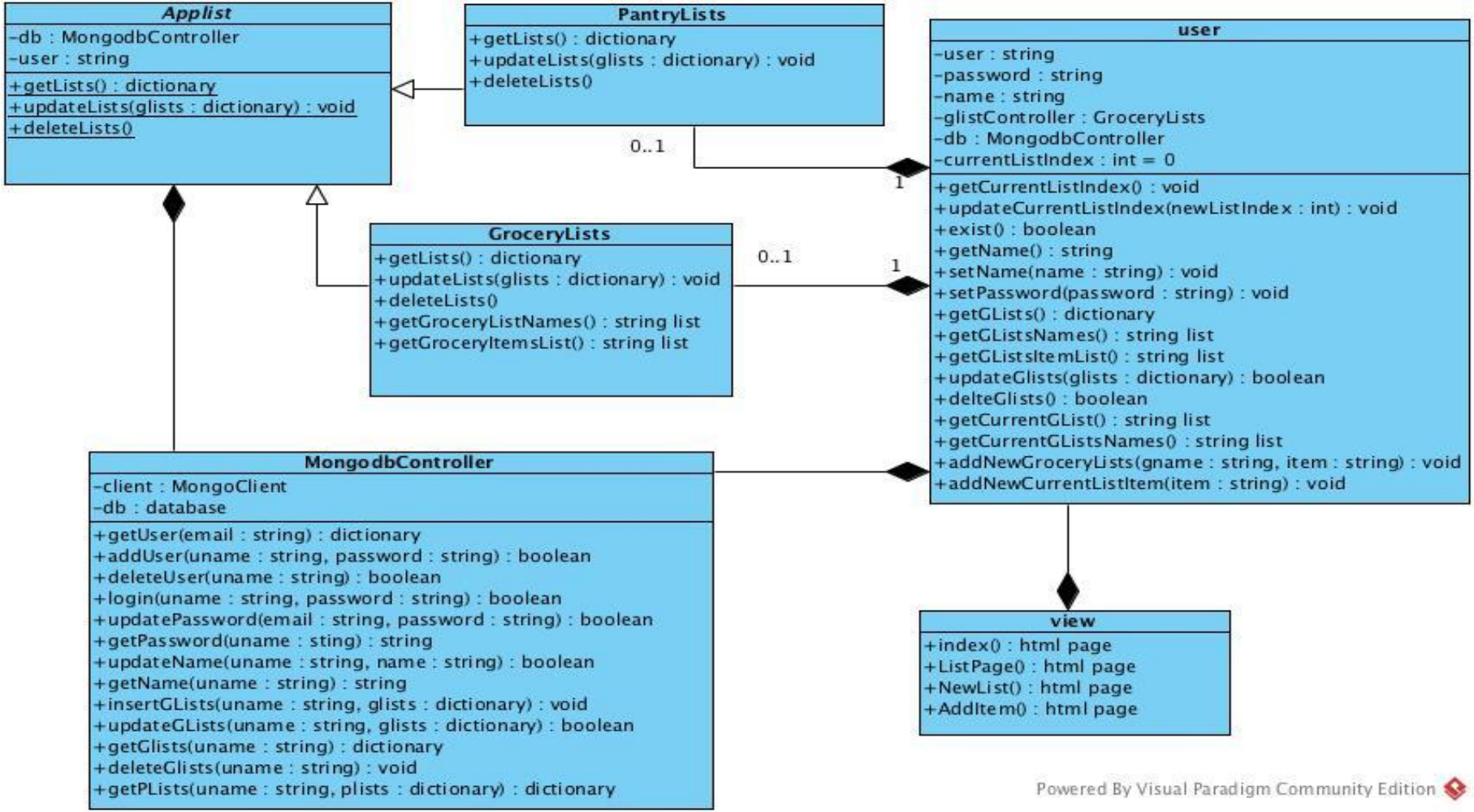
Demo

- Front End
  - Flask
    - Python microframework that handles html web pages
  - Jinja2
    - Provides Python-like expressions for html to dynamically render web pages
  - HTML/CSS/Javascript
    - Makes things look pretty!
- Back End
  - MongoDB
  - Object-oriented Python
- Link - http://grocerylist.herokuapp.com/index.html

Design Structure

- MVC - Model View Controller Architecture
  - Model
    - MongodbController Class
  - Controller
    - User Class - the main communication between the view and the database.
    - Future implementations would implement the observer model to allow multiple users to shopping lists
  - View
    - Flask
    - Future implementation could use the Composite pattern


appLists
-itemList : item
+addItem(item : item) : void
+deleteItem(index : int) : void
+numItems() : int
+moveItemUp(index : int) : int
+moveItemDown(index : int) : int
+getItems(index : int) : item
+updateItem(item : item, index : int) : void
1..*
item
-name : string
-quanity : float
-unit : string
+getName() : string
+getQuanity() : float
+setQuanity(quanity : float) : void
+getUnit() : string
+setUnit(unit : string) : void
databaseController
+addUser(email : string, password : string) : boolean
+loginUser(email : string, password : string) : boolean
+updatePassword(email : string, password : string) : boolean
+updateName(email : string, name : string) : boolean
+updateGroceryLists(email : string, groceryLists : json) : boolean
+updatePantry(email : string, pantry : json) : boolean
+getUser(email : string) : json
groceries
-id : string
+getID() : string
+setID(name : string) : void
pantry
0..*
1
1
1
UITemplate
-user : user
+settingUI() : void
+pantryUI() : void
+mainUI() : void
+logout() : void
user
-email : string
-password : string
-name : string
-grocLists : groceries
-pantryList : pantry
+getName()
+setName(name : string) : void
+setPassword(attribute : string) : void
+getGrocery(index : int) : grocery
+addNewGrocery(grocery : groceries) : int
+addToPantry(pItem : item) : void
+updateGrocery(grocery : groceries, index : int) : void
+editGrocery(index : int) : void
mainUI
+refreshPage() : void
+newGroceryList() : int
+deleteGroceryList() : void
pantryUI
+refreshPage() : void
+moveUp(index : int) : int
+moveDown(index : int) : int
+save() : void
+addItem() : void
+deleteItem() : void
groceryUI
+refreshPage(index : int) : void
+moveUp(index : int) : int
+moveDown(index : int) : int
+save() : void
+addItem() : void
+deleteItem() : void
+editItem(index : int) : void
loginUI
+createUser(email : string, password1 : string, password2 : string) : boolean
+login(email : string, password : string) : boolean
+displayError() : void
settingUI
+changeName() : void
+changePassword() : void
+displayError()() : void
Powered By Visual Paradigm Community Edition


Applist
-db : MongodbController
-user : string
+getLists() : dictionary
+updateLists(glists : dictionary) : void
+deleteLists()
PantryLists
+getLists() : dictionary
+updateLists(glists : dictionary) : void
+deleteLists()
user
-user : string
-password : string
-name : string
-glistController : GroceryLists
-db : MongodbController
-currentListIndex : int = 0
+getCurrentListIndex() : void
+updateCurrentListIndex(newListIndex : int) : void
+exist() : boolean
+getName() : string
+setName(name : string) : void
+setPassword(password : string) : void
+getGLists() : dictionary
+getGListsNames() : string list
+getGListsItemList() : string list
+updateGlists(glists : dictionary) : boolean
+delteGlists() : boolean
+getCurrentGList() : string list
+getCurrentGListsNames() : string list
+addNewGroceryLists(gname : string, item : string) : void
+addNewCurrentListItem(item : string) : void
0..1
1
GroceryLists
+getLists() : dictionary
+updateLists(glists : dictionary) : void
+deleteLists()
+getGroceryListNames() : string list
+getGroceryItemsList() : string list
0..1
1
MongodbController
-client : MongoClient
-db : database
+getUser(email : string) : dictionary
+addUser(uname : string, password : string) : boolean
+deleteUser(uname : string) : boolean
+login(uname : string, password : string) : boolean
+updatePassword(email : string, password : string) : boolean
+getPassword(uname : sting) : string
+updateName(uname : string, name : string) : boolean
+getName(uname : string) : string
+insertGLists(uname : string, glists : dictionary) : void
+updateGLists(uname : string, glists : dictionary) : boolean
+getGlists(uname : string) : dictionary
+deleteGlists(uname : string) : void
+getPLists(uname : string, plists : dictionary) : dictionary
view
+index() : html page
+ListPage() : html page
+NewList() : html page
+AddItem() : html page
Powered By Visual Paradigm Community Edition

Conclusion

- What we accomplished
  - MVC architecture using MongoDB
  - Can add many grocery lists
  - Delete lists
  - Add items to individual lists
- Future Iteration
  - Handling log in of multiple users
  - Delete one or many items from individual lists
  - Implement Settings page
  - Implement Pantry
  - Refine look and feel of site
  - Convert website to mobile app