

Turtl

Software Architecture

Submitted to:

Asst. Prof. Ma. Rowena C. Solamo
Faculty Member
Department of Computer Science
College of Engineering
University of the Philippines, Diliman

Submitted by:
Joan Nicole Balugay
Ram Mangaoang
Brian Sy

In partial fulfillment of Academic Requirements
for the course
CS 191 Software Engineering I
of the
1st Semester, AY 2017-2018

Revision Control

History Revision:

Revision Date	Person Responsible	Version Number	Modification
11/29/17	Brian Sy	1.0	Initial Document; purpose, audience, system name, description, data design classes
11/29/17	Joan Nicole Balugay	1.1	User Interface and Business classes
11/29/17	Ram Mangaoang	1.2	Software architecture diagram

Purpose:

The purpose of this would be to properly lay out the foundation and interactivity of the app to the users. From the processes that make up the program to the user interface, this document will provide insight as to how the program is generally prepared. This shows how different classes would interact with each other and provide interrelatedness between these classes.

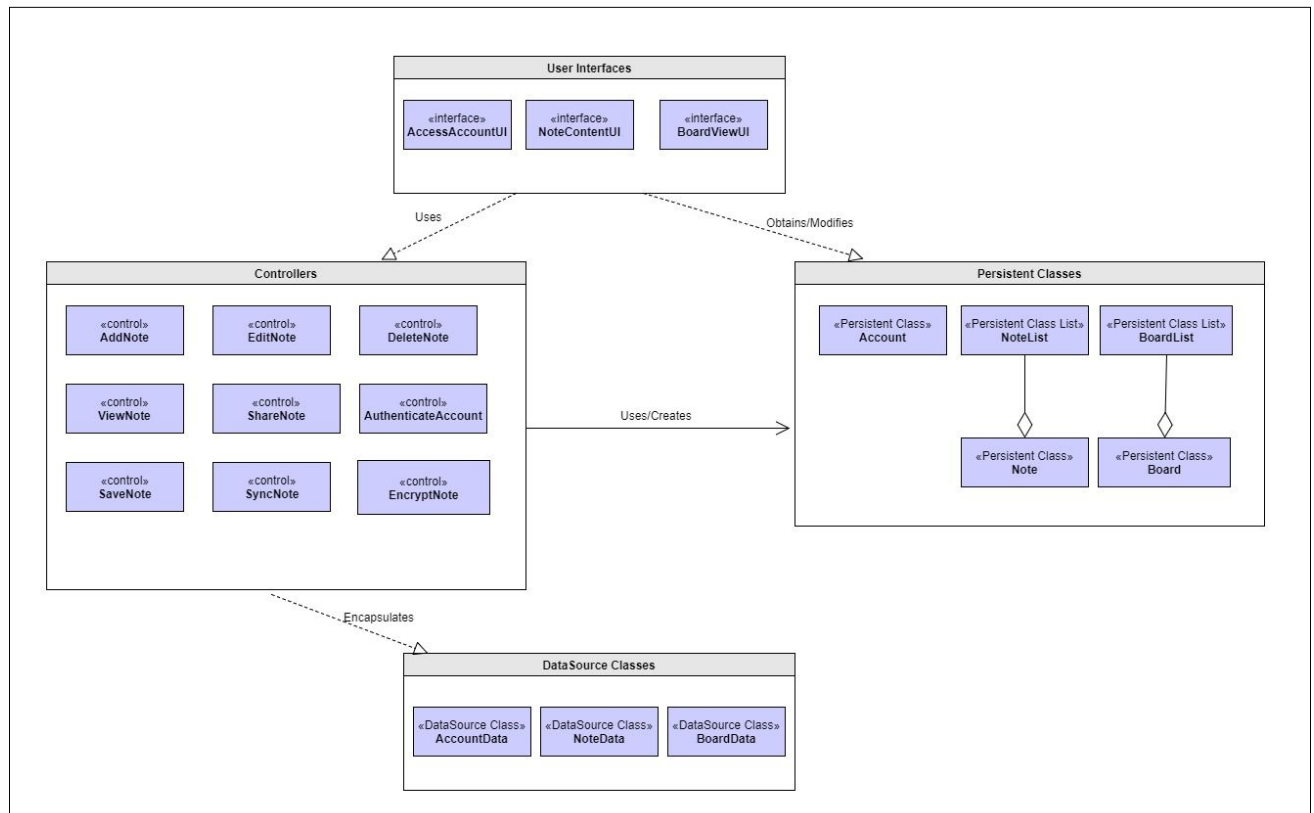
Audience:

The audience would be other programmers who would like to modify the application or businesses that would like to see how the app is being implemented. This way, aligning the application to fine tune towards the app's target audience, or any other future changes, with the rest of its components is more doable and easily implemented.

System Name: Turtl - Sharing, saving, and syncing the specified note

Description: This details how the process of sharing, saving, and syncing would work when seen through the usage of different classes in a structured manner. This way, it would further layout how the usage of these would individually come into play and ultimately turn out with the final product. It creates a systematic view for the project.

Software Architecture Diagram:



User Interface Design Classes:

Screen Name	Description
AccessAccount	Used to be able to access the account. It provides a way for the user to interact with getting into Turtl.
NoteContent	This is used to be able to add contents to the note itself (once the note is already accessed). This would include adding a title, the content itself, saving the note, and sharing the note.
BoardView	This is used to view existing notes contained inside a board.

Business Logic Classes:

Control	Description
ShareNoteController	This will share a board to another user by sending an invite to another user. The board may or may not contain notes. It will take as input a string for the email address of the invitee and another string for the passkey in case the user decides to put a password for the invite.
SaveNoteController	This will save the note and prevent the loss of data. It will take an input of a filename for the note and it will output a save state, which will be represented as a boolean to see if it was saved properly or not. This way, it will be able to save the note that was specified properly.
SyncNoteController	Due to the requirement being that the note has already been saved, it would be put inside a condition that would check this. After which, it will go into syncNote. Essentially, it will synchronize the note from the user's client and the server itself. It needs to take a parameter of the note's filename in order to know what file is being accessed.

Data Design Classes:

Class Name	Description
Account	This will return the account credentials. This can be used for accessing Turtl without the need to always enter the user's account credentials. It can also be used for encryption, as encryption is based upon the user's account
NoteList	This is used to lay out the different notes in a board. This is especially useful for being able to know what notes are within a specific board, and it is beneficial to be able to place them systematically.
BoardList	This will be able to return the contents of the note itself. This transfers the data to be able for the user to properly read and understand the information that is being displayed and transferred from several sources of information in the application.
Note	This will list down all the different boards that are associated with an account. It will be able to lay out what different larger scaled notebooks there are in order to show organization within the system.
Board	This will return the different notes within the board itself. This is closely tied to note list, but this will provide the name of the board, along with a calling to the note list to be able to be more specific regarding the transfer of the different note information.
AccountData	This stores the data of the account. This is a data source class. It would include the current account's credentials for login.
NoteData	This stores the data of the notes. This is a data source class. It would include the current board associated with the note, the type of note, and the filename associated with it.
BoardData	This stores the data of the board. This is a data source class. It would include the the name of the board and the number of notes associated with the board.