Turtl
Analysis Model

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***
10/24/2017	Brian Sy	1.0	Initial Document

Purpose:

The purpose of this document is to be able to fully detail how different parts of the software would interact with each other and what kind of specific functionalities each one will exhibit. By doing so, it can clear up confusion and ambiguity with the terminology and functions.

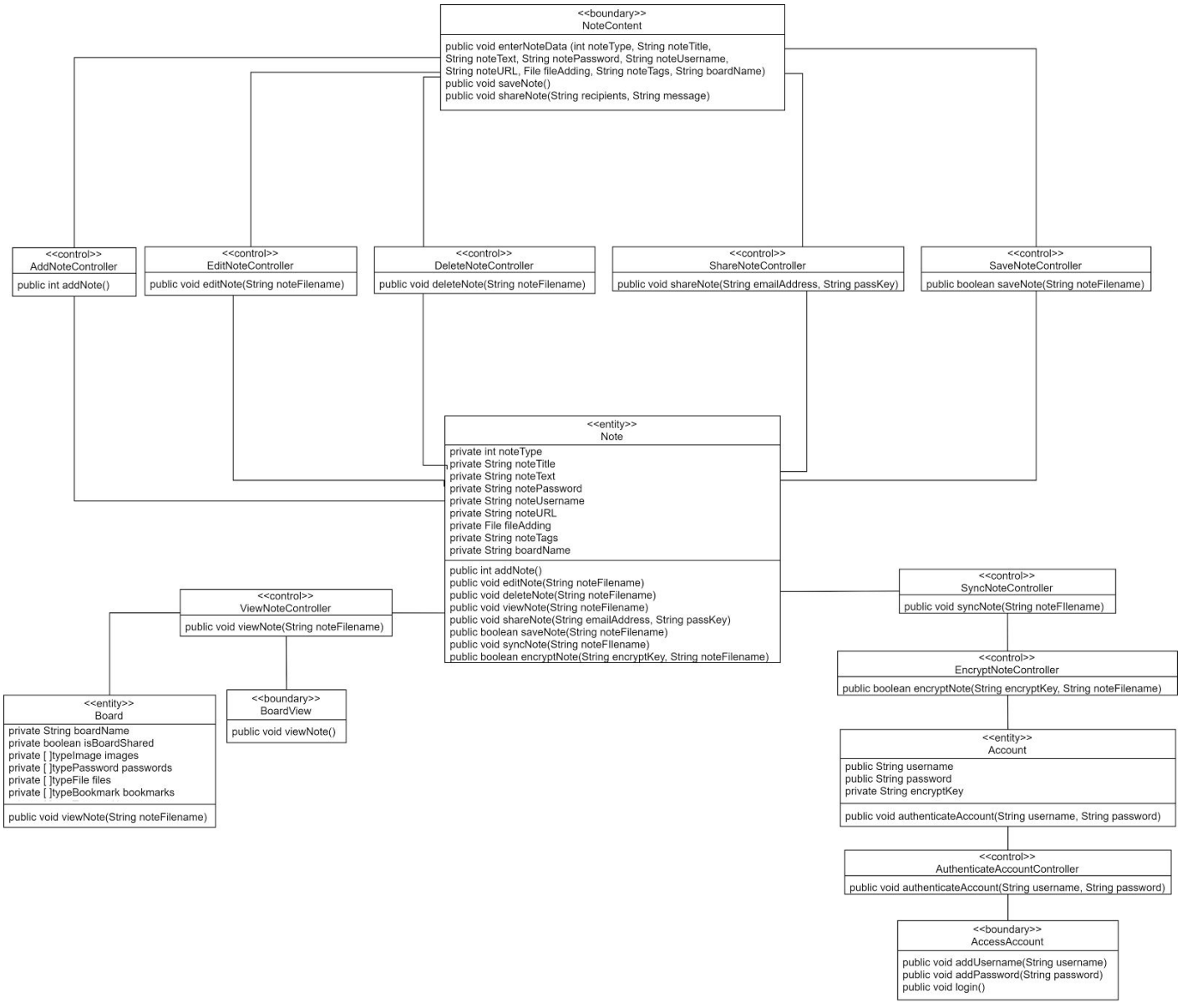
Audience:

The audience of the document would be those interested to see how the program would work on a lower level. This would include other programmers, businesses who would like to see how the inner workings of the software would work and how the content ties up together.

System Name: Turtl Note Maintenance and Saving System

Description: The system would detail how maintenance within the user and the server would work. It would detail how adding or editing a note would look like, for example, and how saving would also tie up with the server and ultimately back to the user's client. This way, it provides a clearer picture as to how things would tie up in the end.

Class Diagram:



Boundary Classes:

Class Name	Description
AccessAccount	Used to be able to access the account. It provides a way for the user to interact with getting into Turtl. public void addUsername(String username) public void addPassword(String password) public void login()
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. public void enterNoteData (int noteType, String noteTitle, String noteText, String notePassword, String noteUsername, String noteURL, File fileAdding, String noteTags, String boardName) public void saveNote() public void shareNote(String recipients, String message)
BoardView	This is used to view existing notes contained inside a board. public void viewNote()

Control Classes:

Class Name	Description
AddNoteController	Adding a note, creates a new instance of a note. After creating a new instance, the user will choose whether to add a password, file, image, bookmark or text notes. This will return an integer which will be used by the edit function to determine what kind of note will be edited. public int addNote()
EditNoteController	This will edit a newly created note or an existing note. This will take an input of the note filename to be edited. It will check first what type of note will be edited. After confirming, the note will be edited based on its type. The note is edited by changing its variables. After editing, the note will be saved. public void editNote(String noteFilename)
DeleteNoteController	This will delete a note and its data from the server. It will take an input of the filename of the note, which will be deleted from the server. public void deleteNote(String noteFilename)
ViewNoteController	This will let the user view an existing note. It will take as input the filename of the note, which will let the data be accessed from the server. public void viewNote(String noteFilename)
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. public void shareNote(String emailAddress, String passKey)
AuthenticateAccountController	This will allow the user to log in to the app. It will take as input two strings: one for the username and another for the password. However, the server will first check if the information inputted by the user is associated with an existing account before allowing the user to log in. public void authenticateAccount(String username, String password)
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. public boolean saveNote(String noteFilename)
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. public void syncNote(String noteFIlename)
EncryptNoteController	Encryption will return a boolean that will say whether the note has been crypted or not. It uses the encrypt key from when an account is created

	and uses this in order to encrypt the files for safety. It also uses the noteFilename that was already synchronized (a condition for this can be made) and works with that note. After this, there would be a flag that can tell other parts of the software that the note was encrypted, if needed. public boolean encryptNote(String encryptKey, String noteFilename)

Entity Classes:

Class Name	Description
Account	An account would need to have a username, password, and an encryptKey that was provided by Turtl account generation. This assumes that the user has already made an account on Turtl's website. This is merely for authentication and login. private String username private String password private String encryptKey
Note	A note can be classified into password, image, file, bookmark, and text note which will have a corresponding type number for each. A note can have a title, a note text, a password, a username, a URL, a file, tags and board location. private int noteType private String noteTitle private String noteText private String notePassword private String noteUsername private String noteURL private File fileAdding private String noteTags private String boardName
Board	A board contains notes and can be a combination of any of its classifications: password, image, file, bookmark, and text note. It will contain a name and an array for each note type for storage. It also contains a boolean variable for determining if the board is shared with another user or not. private String boardName private boolean isBoardShared private [ ]typeImage images private [ ]typePassword passwords private [ ]typeFile files private [ ]typeBookmark bookmarks private [ ]typeText textNotes