course: **CS446** team: **BB8** project: **WAThub**

project status report

progress...

This is the first status report of the project. First of all, let's summarize the progress we have made since last month till now. If we refer to the estimated breakdown of time management for major activities of the project provided in the original proposal, we can certainly say that we have met the deadlines almost exactly with a few not yet done items. Let's go through each one of them briefly below:

- project setup

This item has been done right after the submission of the proposal. This includes git repo setup, android studio installation and project creation, asana project tracking, parse integration and slack channel setup.

- architectural design

The overall and high level architecture of the project has been discussed in the weekly meeting of the teamwork and detailed design of the major activities has been determined using board mockups, diagrams and design patterns.

- database design

In the next meeting, design of the project database has been completed using complex uml diagrams. Parse has been accepted to host the project database and provide server side implementation to allow us to focus on the android app itself and to ease the process of data management and exchange between database and the application.

- database implementation

This item as also indicated in the original proposal, is supposed to be done during the course of the project as although its first defined structure has been implemented, it may be modified and/or extended as application progresses and further development decisions require so.

- login/logout

This major activity has been completely implemented and material design principles are to be applied at the end of the project to this item. This feature now allows users to login, sign up with their UW emails verified and reset password by following reset instructions sent to their emails and eventually log out of the application as well.

- profile creation

The basic structure of this activity is almost done and few extensions are to be applied during this week to make it fully available for the upcoming demo. This activity includes the picture of and information about the user.

- event post

This component which is also going to be a parent and skeleton activity for other posts as well, is essentially done with few adjustments expected to be finished this week and be ready for the demo. Posts are going to be simple Google cards which can be slided either way and are quite flexible to manipulate between other objects.

- group study post and book exhange post

These items are still in progress and are expected to be completed before the upcoming demo. However, basic framework for these posts is already finished and only specific parts of implementation are in development process.

problems and solutions

Other than few technical issues with project setup and database design, one particular problem can be noted to emphasize importance and significance of design patterns in software projects and this should be considered as a design problem as well. This issue has been identified when using Parse subclasses inside java packages. Fortunately, Parse allows developers to define special classes that are associated with specific data classes in the backend and provides object oriented interface with getter and setter methods to access backend database more easily. It extends ParseObjects and hence, has all of the features of the ParseObjects. Instead of hardcoding database class names and fields and accessing them all over the code by making it hard to modify later on, developer can just use these wrapper classes to create new instances of backend classes and save them easily using provided ParseObject methods. However, when it comes to retrieving data from the existing objects in the database, it is not clear how to use the same wrapper class to get fields of that class as it is a child of ParseObject, but ParseObject does not have those specific get, set methods. This introduces the problem downcasting which is not possible sometimes at run time and most of the times at compile time depending on the types of objects they refer to. In this specific scenario, although this downcasting operation passes compiler checking, but it fails run time checking as retrieved ParseObjects do not point to child wrapper objects.

One obvious solution would be to just use the ParseObject methods directly, but this would make the code again hard to modify later on at multiple locations. This immediately signals that there is a design interface problem and it should hence be solved using one of design patterns. The one that solves this problem is the Adapter Pattern with object composition. This pattern allows us to define another ParseObject inside our wrapper class and do all the get/set operations on this object as well. This also requires to create 2 overloading constructors where one accepts another ParseObject to operate on and the default one with no arguments that maintains our old feature of creating new instances of ParseObjects with all the specific get/set methods.

new activities

- material design

As a new activity, we aim to materialize our project with elegant, simple and colorful material design principles and components at the end of the functional implementation of project activities. We estimate this to be done towards the end of the project.

end of the status report.