Alvin Mao (andrew id: asmao)
Kevin Dou (andrew id: khd)
Product Backlog
Due: 10/30/14

Team: ASM Simulator (team name in github: Team101) Product Backlog

Profile Features:

Profile features cover the account information. This covers 5 areas: registration, login/out, edit/view profiles, files view, and search. There are an additional 3 areas we would like to cover if we have time: account upgrade, file deletion after long periods of inactivity, and various share on Facebook/Twitter etc… tags. The objectives will be marked with a "FUTURE OBJECTIVE" string.

1) Register
    a) The user fills in the required fields: email, first name, last name, password, confirm password.
    b) Upon completion, this will send a confirmation email to the user and the account will only be activated upon confirmation of that email.
    c) FUTURE OBJECTIVE: In the registration the user can also select the account type (premium/standard) she wants. See the step 6).
2) Login/Logout
    a) When user logs out and hits the back button they do not get redirected to the previously logged in page.
3) Edit/View profile
    a) User can edit their basic registration information here.
    b) The user can view how much space there is remaining and the net space she has been allocated.
    c) FUTURE OBJECTIVE: The user can upgrade their account to premium. See step 6)
4) View Files in Home
    a) The user can view all of their files in their home page. The user can filter the files by owner, date, upload and more. This is page analogous to the home page of google drive/dropbox where the user can view all the files in their drive.
    b) Each individual file shown displays the title, version number, and a sample of the file description. The user can click on a more details of the file and display the complete description.
    c) When the user clicks the open file button, the user will be brought to the IDE page where the user can modify and simulate their code.
    d) The user can create a new file or upload an existing file.
5) Search for AsmFiles
    a) The user can choose various filters to search for various files. The user can search by creator, title, and a generic search that searches for keywords in description as well as the title.

    b) After performing the search, a list of the files will be displayed and the user can then download such files. The user can perform similar operations on files as they can in step 4b).
    c) The user can download the file which can then be accessed in 4) and in turn opened in the IDE page.
    d) FUTURE OBJECTIVE: The user is limited in the number of downloads if they only have a standard account. Premium users have unlimited downloads.

FUTURE OBJECTIVES:

6) Account Upgrade
    a) There are two modes: standard and premium. In the standard account the user is limited to 5 MB and can only download 5 files. In the premium account the user can download and has 1GB of space.
    b) The user can also purchase additional storage at some predetermined price.
    c) Users will be alerted by email when they hit their space limitations.
7) File Deletion After Some Time
    a) If the user hasn't logged in for some time, they will have their files deleted after some period of inactivity.
    b) The user will receive an email notification about 7a).
    c) The file deletion protection will be provided to premium members.
8) Share on Twitter/Facebook etc…
    a) Share selected files/information about the site.

IDE Features

IDE Features revolve exclusively around the IDE page. We will be creating a very simple 16 bit ISA with an even simpler architecture. The ISA and the architecture will be covered in a separate ISA documentation which will be created by the end of the first sprint (ISAdoc.pdf).
The IDE page can be divided into the following sections: editor, console, main memory, registers, and run/compile/debug/save/stop/step commands.

1) Editor
    a) Here, the user can type in their code.
    b) The left side will be lined with line numbers.
    c) The user can set a breakpoint by clicking the line number of a valid instruction.
2) Console
    a) Displays the errors or returns a success message upon compilation/run/debug.
3) Main Memory
    a) Here, the user can view their program starting from 0 of main memory and if they scroll to the bottom they can see the stack. The PC in main memory will be highlighted in blue and the SP will be in red. When both overlap it will be in purple (you're probably in trouble if this is happening).
    b) There will also be a SP and a PC button that allows the user to jump back and forth in the main memory view.
4) Registers

    a) Displays the 4 basic registers, PC and SP, and the condition code registers the user can view.

5) Program Options
    a) Run - Interprets bytecode and runs to completion and ignores breakpoints.
    b) Stop - stops the execution and clears main memory and registers.
    c) Compile - verifies that all the instructions in the editor are syntactically correct and pumps out the resulting bytecode to insert into main memory.
    d) Debug - enters debug mode where the user can step through the code/run to breakpoints.
    e) Step - executes one assembly instruction.
    f) Save - saves current code
    g) Close - close current file and if changes have been made as user if they want to exit without saving. Upon successful close the user returns to home.

FUTURE OBJECTIVE

6) Watchpoint Implementation
    a) The user can set watchpoints at certain locations in memory that trigger whenever the PC/SP touch that point in memory.
7) Editor Improvements
    a) Syntax highlighting
    b) FIXME, TODO, XXX markers that will be displayed on the right hand side of the editor when typed.