**CS 422/522**

**Presentation Flow**

**By: Ryan Helms, William Qiu, Nikhar Ramlakhan, Abie Safdie and Caleb Sutherland**

**PRESENTATION REQUIREMENTS:**

A. Explain what you did:

* Briefly describe the system's user perspective and Concept of Operations. (10 seconds)
  + Nikhar
* Demonstrate essential system functionality without the login process. (4 minute)
  + Team and Nikhar to narrate
* Provide a quick overview of the utilized technologies. (30 seconds)
  + Nikhar

B. Explain how you did it:

* Translate requirements into software design, showcasing the process and evolution through diagrams, with one design insight. (30 seconds)
  + William
* Integrate and test systems, discussing successes, challenges, and one testing lesson learned. (30 seconds)
  + Caleb
* Manage the team and experience, highlighting successes, challenges, and three key teamwork learnings. (1 minute)
  + Abie
  + Ryan

**0m 00s ~ 1m 10s**

1. **Introduction (1 minute 10 seconds)**
   * Group introduces themselves (30 seconds) - All
   * Provide overview of project (10 seconds) – Nikhar
   * Provide overview of the technology used (30 seconds) - Nikhar

**1m 00s ~ 1m 40s**

1. **Server set up – Abie** 
   1. Ix dev ssh into
   2. MySQL as a chosen platform
   3. Setting up a connection
   4. Mysqlctl preinstall ( to save time)

**1m 40s ~ 2m 40s**

1. **Admin set up – William**
   1. Default server
   2. Error message
   3. Why is the server required to be set up before usage?
      1. So it doesn’t rely on predeterminate server
      2. By default, it is connect to one of our ix-dev.

**2m 40s ~ 3m 25s**

1. **PDF page – Ryan**
   1. What happens in backend.
      1. Runs for loop over PDFs folder, constructs entries in database for each PDF. Notes are unique for each user.
   2. PDFs loaded from the source directory
   3. It generates a button for PDF

**3m 25s ~ 4m 25s**

1. **Annotation page – Caleb**
   1. Show how you can SQ3R
   2. Hide / show notes
   3. Note hierarchy

**4m 25s ~ 5m 10s**

1. **Translating SRS into SDS – William**
   * Initial SRS
   * Misunderstandings
   * Understand the SRS
   * Define the software architecture
   * Breakdown the software modules
   * Detailed design
   * Interface
   * **Include model diagrams**
   * **State at least one thing that your group learned about software design.**

**5m 10s – 5m 40s**

1. **Testing – Caleb**
   * **State at least one thing that your group learned about software testing.**
   * Different systems and versions of modules (Python etc)
   * Determining whose end the issue is on
   * Prepare the integration environment.
   * Define test case
   * Challenge: Environment Discrepancies(different environment led to unexpected behaviors(bugs shown on others computer))
   * Testing lesson learned: **early and frequent testing**.
   * Process: Each person tests individually and then works together to integrate them together.

**5m 40 s – 6m 40s**

1. **Teamwork – Ryan and Abie**
   * Git hub and working with a multi-person repo and updating logs. General software dev communication.
   * Hornof quote for a topic on miscommunication.
     1. Asking for help is okay.
   * Planning and dividing up tasks are helpful for everyone involved
   * Talk about team dynamics
   * General reflections
   * Working as a software engineer
   * Update logs
   * Using git