CyNote

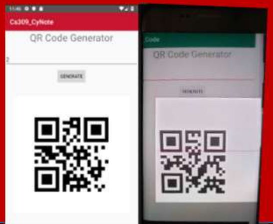
COM S 309 SPR 2019

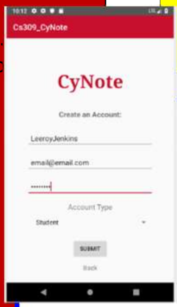
Project Description:

An educational tool similar to canvas. In addition to a class and lecture hierarchy for students, we provide a live, collaborative note taking environment. Students may also use our shoutout tool to anonymously ask the professor questions during lecture. Students can also share images of their handwritten notes. These notes can be voted on based upon quality at the discretion of a TA or professor. There is also a QR code creator/scanner for professors to give students a QR code so students may scan it and automatically be added to the given class.

Users:

Students
Professors
TAs
Administrators





Create an Account:

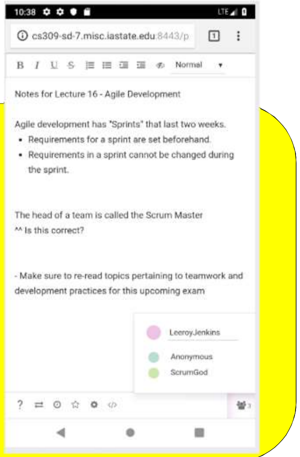
LeeroyJenkins

email@email.com

Account Type: Student

User Interfaces

- Live Collaborative Note Taking
- Shoutout Live Chat
- Textbook Sharing
- Notes Sharing and Voting
- Class List
- Lecture List
- File Upload



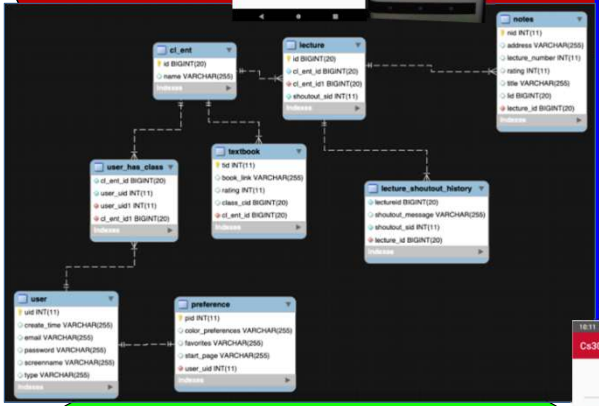
Notes for Lecture 16 - Agile Development

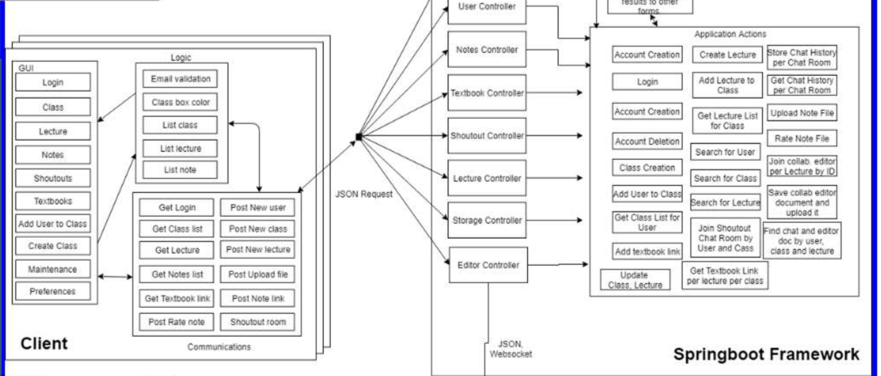
Agile development has "Sprints" that last two weeks.
- Requirements for a sprint are set beforehand.
- Requirements in a sprint cannot be changed during the sprint.

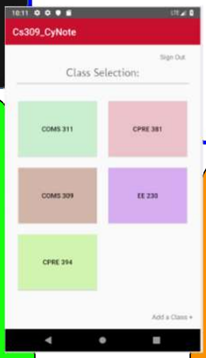
The head of a team is called the Scrum Master
^^ Is this correct?

- Make sure to re-read topics pertaining to teamwork and development practices for this upcoming exam

LeeroyJenkins
Anonymous
ScrumGod







Class Selection:

Module Interfaces:

Back-End API Endpoints and Types (generalized) -

User -
Represents an account in the system, has access privileges ranging from Amin → Professor → TA → Student.
Each *User* is linked to a list of *Class Entities* that they are currently enrolled in.
Functional endpoints to create, deactivate, and modify a given *User*.

Class Entity -
Represents a school course in the system. Each *Class Entity* is linked to a list of *Lectures*.
Functional endpoints to create, deactivate, and modify a given *Class Entity*.

Lecture -
Represents one day's lecture for a course in the system.
Each *Lecture* is linked to its own *ShoutOut* chat, *Live Notes* room, and a list of of *Static Notes* in order of rating.
Functional endpoints to create, deactivate, and modify a given *Lecture*.

ShoutOut -
Represents the live chat between Students, TAs, and Professors during one day of class.
ShoutOut rooms are only open for the duration of the physical lecture, plus a small amount of leeway, after which they will be closed to further messages by Students but still viewable in the app.
Functional endpoints to create, deactivate, and modify a given *ShoutOut*.

Live Notes -
Represents the live, collaborative note-taking efforts of any participating students for a specific lecture, viewable and editable at any time.
Functional endpoints to create, deactivate, and modify a given *Live Notes* room.

Static Note -
Represents a single *User*'s uploaded notes for a specific lecture, able to be upvoted and downloaded by anyone in the class.
Functional endpoints to create, deactivate, and modify a given *Static Note*.

Front-End API Calls (generalized)* -

volleyPost(...), volleyGet(...), and volleyGetArray(...)
Generic requests to a provided server endpoint, used in most locations to communicate with the server.

getUserNoClassList(...) with getClassList(...)
Returns a completed User, used to populate the *Class Selection* screen and is the starting point for a login.

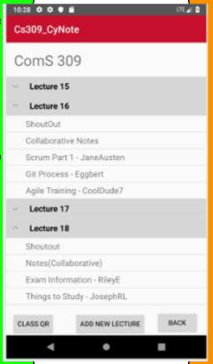
getModuleList(...)
Updates a given Class with its respective list of lectures/modules, used to populate the *Lecture Selection* screen and provide updated information for processes.

getShoutOutHistory(...) alongside getNoteList(...)
Updates a given Lecture with its ShoutOut message history and its list of Static Notes respectively, used to populate dropdown menus in the *Lecture Selection* screen and provide updated information for processes.

***Note: Front-End uses most of the same object types as Back-End above, but their descriptions are excluded for brevity.**

Design Decisions:

- •Database for instances of Live Notes
- •Database for ShoutOut chat history
- •Real-time refresh for Live Notes and ShoutOut
- •Class Selection and Module/Lecture Selection auto population
- •QR code scanner and generator for ease of access
- •Break API into simplified function calls passed generic callback interfaces
- •Offload calculations to server-side, keep client-side lightweight

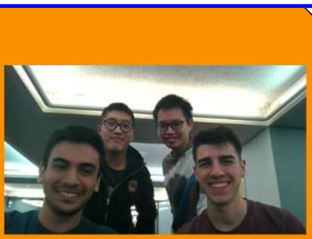


ComS 309

Lecture 15
Lecture 16
ShoutOut
Collaborative Notes
Scrum Part 1 - JaneAusten
Git Process - Eggbert
Agile Training - CoolDude7
Lecture 17
Lecture 18
Shoutout
Notes(Collaborative)
Exam Information - RileyE
Things to Study - JosephRL

Team Info:

Team SD_7
Marc Issac – ComS(Jr)
Sean Gordon – CPRE (Soph)
ZheMing Fan – ME (Jr)
Shen Chen – ME (Sr)



What went wrong:

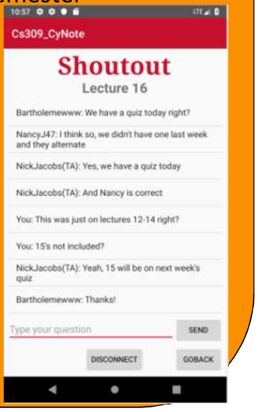
- Lack of communication between frontend and backend at the beginning of the semester
- Had to change data structure mid-project

What went right:

- Completed project far exceeding expectations
- Great teamwork

Lessons Learned:

- Test the project often
- Construct data structure carefully



Bartholemewww: We have a quiz today right?

NancyJ47: I think so, we didn't have one last week and they alternate

NickJacobs(TA): Yes, we have a quiz today

NickJacobs(TA): And Nancy is correct

You: This was just on lectures 12-14 right?

You: 15's not included?

NickJacobs(TA): Yeah, 15 will be on next week's quiz

Bartholemewww: Thanks!