CMPS 480 Senior Project - Module 1: Defining the Prototype

Mark Voortman, Ph.D.

Course Introduction

- This is an exciting course where you get to **apply all skills you have learned throughout the Applied Computer Science program**. Hands on is a large part of the course and you will be working on a semester long project.

Course Description

- **CMPS 480 Senior Project**
- **In this course the student will demonstrate their mastery of material undertaken in coursework by selecting and creating a programming solution to a significant business application.** The group will work together to construct their solution and present a working model of their problem to the class. This project is designed to give the student a hands-on demonstration of their coursework suitable for a portfolio of accomplishments. *Prerequisites: All I.T. Core Classes. 3 credits.*

Course Objectives

- Upon successful completion of this course, you should be able to:
  - Define business and user requirements for a software application
  - Develop and deploy a software application including a front end and back end component
  - Apply a business analytics approach as a component of the software application
  - Document the development process including security considerations
  - Collaborate in different roles as part of a team environment
  - Present results at various stages of development in a professional manner

Course Modules

- Module 0: Course Design
- ***Module 1: Defining the Prototype***
- Module 2: User Interface
- Module 3: Database
- Module 4: Web Server
- Module 5: Business Analytics
- Module 6: Finding Customers
- Module 7: Refinements
- Module 8: Releasing the Prototype

Module 1

- **Defining a Prototype**
- Part A
  - Form groups
  - Business model canvas
- Part B
  - Requirements
  - Architecture
  - APIs
  - Security

Objectives

- Upon successful completion of this module, you should be able to:
  - …

PART A

Introductions

- Instructor introduction
- Please tell us about your **major, background, interests, and what you hope to gain from this course**
  - Online: See Discussion Board or post to Element
  - On-Ground: Go around the classroom

Course Design

- **8 modules in total:** for both online and on-ground
- Each module is divided in **Part A and B**
  - Online is 8 weeks: Complete **one module per week**
  - On-ground is 15 weeks: Complete **one module every two weeks**
  - The **final module** is completed in the **final week** in both cases
- For more info see *Module 0: Course Design*

Syllabus

- See Canvas course shell for **syllabus**
- Make sure to **read carefully**
- Most important is to work through **these slides** every week
- Submit homework **on time** (start early!)

Using (AI) Tools

- **Read the policies** on how you are allowed to use **AI tools**
  - This is **IMPORTANT**
  - More to follow

Required Books and Materials

- There is **no required textbook** for this course
- Rely on **publicly available** resources and tools
- Utilize **open educational resources**
- Necessary materials are always **linked** from slides

Extra Credit

- Make sure you are **engaged** in class and with the content
- To be successful, **reach out for help** when needed
  - **Do not wait** until it is too late
  - It will only make things **more difficult**
- Therefore, **extra credit** is typically awarded for active participation

How to Ask Questions

- Related to reaching out for help, it is important to **ask questions properly**
- *"it doesn't work"*
  - Too often instructors receive one liners from students without any context
  - More effort is expected than sending a text message
- Instructions on how to ask questions
  - This page contains some good pointers that should be followed

How to be Successful

- **There are simple steps one can follow to be successful but that is typically not easy**
  - Most students know what to do to be successful
  - But it is not (always) easy to execute on this
- Instructions on how to be successful
  - This was compiled from class discussions with students
- Key takeaway: **form habits** to be successful long term
  - Will be revisited

Build a Portfolio

- Start creating a **portfolio** for when eventually applying for a **job**
  - This can help make your resume look a lot better
- Instructions on how to build a portfolio
  - Some ideas are shared on this page

Best Way to Contact

- Preferred
  - **Element chat** (see next slide)
- Alternatively
  - Email
  - Canvas messaging

Using Matrix/Element

- Built on the **Matrix protocol**
  - **Open** standard and communication protocol
- Can use any client but **Element** is recommended
  - Follow the instructions to use Element
- You will be added to the **Point Park University Space**
  - Several **rooms** are available to join
- Benefits
  - **Realtime** interaction for asking questions (as opposed to email)
  - It is **federated** like email
  - Receive **updates** about the Applied Computer Science program
  - Learn about internship and co-op **opportunities**
  - **Stay in touch** even after you graduate

Additional Resources

- The Applied Computer Science website provides many resources
- Continuously updated with new information, tutorials, etc.
- For example
  - Tutorials (for more advanced classes)
  - Courses (including descriptions and objectives)
  - Etc.

Utilizing the Jump Box

- The purpose of the jump box is to be able to connect to your FreeBSD jail
- Unfortunately we are not allowed to connect to the jails directly anymore
- Follow the instructions to use the jump box
- Make sure access was provisioned (ask your instructor)
- You can use PuTTY to connect to your FreeBSD jail over SSH
- You can use FileZilla to upload files to your FreeBSD jail

Using Your FreeBSD Jail

- Make sure a jail was provisioned to you (ask your instructor if unsure)
- Use PuTTY to connect to your jail
- Save the settings to prevent reentering credentials every time
- Watch the second half of this video for the steps

Possible Missing Background

- Depending on your individual course sequence, you may lack some background that is useful in this course
- For example, normally one would take the Senior Project after Advanced Programming but there are some exceptions to this rule for this group
- Some of you have more experience, so you can help out the others. I will also help along the way but you have to take the initiative.
- Especially the back end (server side) may be tricky
  - Interact with the frontend(JSON)
  - Interact with the MySQL database (SQL)
  - More on this in a later module …

What Will You Work On?

- You will work on a semester long group project
- It is almost a startup setting
  - You have to write something functional quickly
  - There are few resources–time is limited
  - You may need to learn some new things on the way
  - You should try out your prototype on actual customers
  - Ideally, the idea should be commercially viable (make money)
  - A good example is the Pioneer Pantry website
- You should come up with your own project but it has to be approved by me

Create Groups/Teams

- You should split into groups of approximately 4 students
- Each team should pick a team leader
  - Responsible for communicating problems to me
- Each group should pick their own project
- There will be a competitive element
  - Who does a better job?
  - You can also learn from each other

Business Model Canvas

- Business plan in simplified and visual form
  - In my opinion a much better version of a business plan
- Read the complete Wikipedia page
  - https://en.wikipedia.org/wiki/Business_Model_Canvas
- Work through this tutorial
  - https://www.alexandercowan.com/business-model-canvas-templates/
- You should create an online business model canvas here
  - https://canvanizer.com/new/business-model-canvas
- You may not have all the information yet, in that case just use your best possible guess (you can fix it later)

Set Up Tools

- This includes
  - GitHub
  - Git
  - Matrix/Element
- Everything should go into GitHub
  - This includes code, but also all the documentation such as the business model canvas
  - Do this from day 1
- These tools were covered in Advanced Programming and other previous course work

Things to Cover

- Create teams
- Pick a team leader
- Pick a project
- Fill out business model canvas
- Set up GitHub, Git, and Matrix/Element
- Think from the perspective of the customer
  - What questions would you ask a potential customer?
  - Try to find out what the pain points are
  - I will play the customer throughout the course
- Next week
  - Refine business model canvas
  - Understand enough to build a first prototype
    - What are the requirements?
    - What features should be included? Etc.

Assignment

- See Canvas for details

PART B

UML

- This is also covered in CMPS 262
- **Unified Modeling Language**
- Should **remember** this term
  - Used often to specify program **designs**
- Consists of many types of diagrams but we will only use a **few**
- Study the following articles to get an impression:
  - https://www.smartdraw.com/uml-diagram/
  - http://creately.com/blog/diagrams/uml-diagram-types-examples/

Topics

- Requirements
- Architecture
- APIs
- Security

Requirements

- Create 3 (important) use cases (part of UML) for the user of the prototype
  - I.e., how the user would use with the system
  - Study https://en.wikipedia.org/wiki/Use_case
- A use case diagram shows the user and the use case
- Follow the structure under Examples on Wikipedia
- See next slide for an example

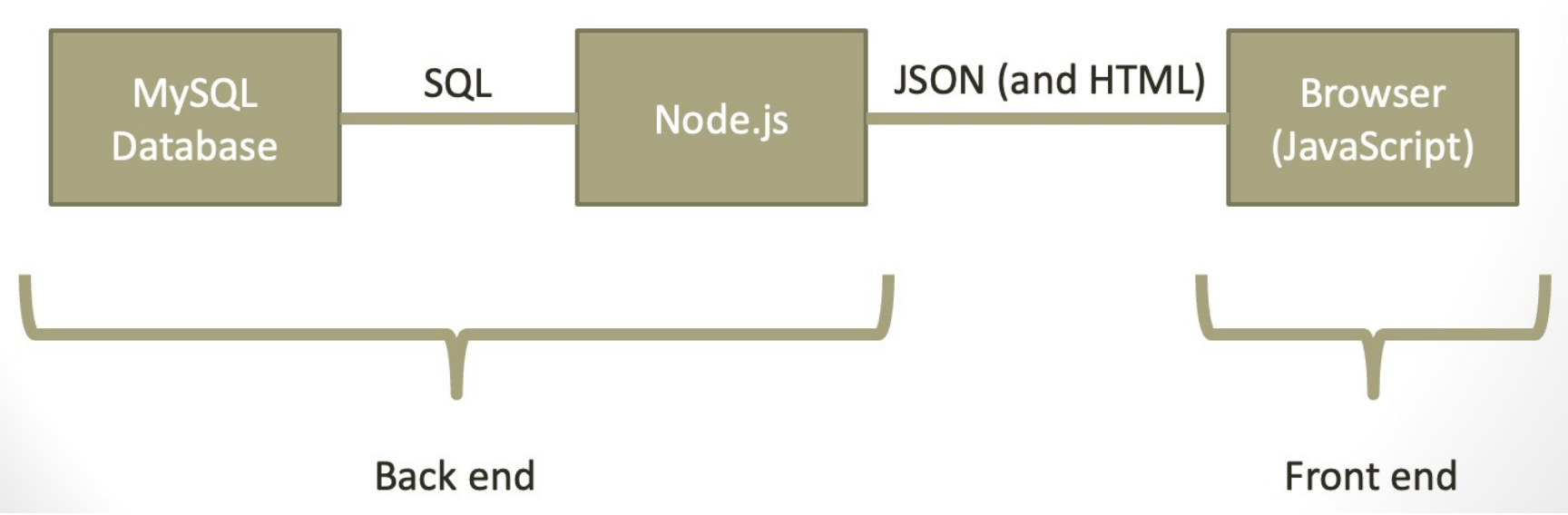
Use Case Example

- **Use Case**: Update the repository
- **Primary Actor**: Administrator
- **Scope**: The inventory system
- **Level**: User goal
- **Brief**: The user logs into the system and updates the repository with a newly arrived order of food
- **Stakeholders**: Administrator, Food Pantry management
- **Postconditions**: Database is updated
- **Preconditions**: Administrator has account
- **Triggers**: Food order arrives
- **Basic flow**: 1. Log in 2. Enter a new line for each food 3. Press the process button 4. The database is updated
- **Extensions**: …

Architecture

- See https://en.wikipedia.org/wiki/Software_architecture
- High level overview of the system
- What are the components?
- What are the interactions?
- Draw a box and line diagram (see next slide for example)

Box and Line Diagram Example



API

- Application Programming Interface
- What data will be sent between Node.js and the Browser?
- See example using JSON for data exchange on next slides
  - See here for an introduction to JSON

API JSON Example Input

```
{
  "action": "create_account",
  "name": "Mark Voortman",
  "email": "mvoortman@pointpark.edu",
  "password": "some_password"
  }
```

API JSON Example Output

```
{
  "success": true,
  "message": "Account was created."
}
or
{
  "success": false,
  "message": "Email already in use."
  }
```

Security

- How do you make sure the system is secure and data safe?
- Document how you will achieve that
- What data is sensitive?
  - See:
    - https://en.wikipedia.org/wiki/Personally_identifiable_information
- How will you protect it?
- For example, use HTTPS for logins to not leak passwords

Assignment

- See Canvas for details

THANK YOU!