WhootChat

CEN 4010 Principles of Software Engineering

Milestone 3 Project Proposal and High-level description

Team 10
Low Orbit Programmers

Matthew Vohland - mvohland2017@fau.edu

Darion Chong -dchong2019@fau.edu

Redrick Horton-Schittko -rhortonschit2017@fau.edu

Trinity Carnegie - tcarnegie2017@fau.edu

Aicsa Machado - machadoa2016@fau.edu

Florida Atlantic University
Dr. Shihong Huang
shihong@fau.edu

April 6, 2021

Done

1) Milestone 3 document – an expanded version of Milestone 1

2) A vertical software prototype

Part 1: Milestone 3 document:

Milestone 3 has to be reasonably consistent with Milestone 1 and instructors' feedback but it can also differ from Milestone 1 based on what you discover and develop in your design process in spirit of iterative software engineering process and based on the feedback you get.

The difference between M1 and M3 DO NOT need to be edited in Milestone 1 document which remains frozen. You should start with Milestone 3 only after you have incorporated instructors' feedback on Milestone 1. Milestone 3 document is a separate document from Milestone 1.

Part2: Vertical software prototype

In addition to the Milestone 3 document, the team will create a "vertical software prototype" to test the infrastructure and chosen frameworks and to jumpstart the coding effort. The vertical prototype is the code that exercises full deployment stack from browser, via middleware, to DB and back-end, including your chosen framework. It has to be deployed from team account, in the same way that the final product will be deployed. For example, it shall allow one to enter a search term in the browser, then get a response form the DB and render it back on the browser. GUI for this can be simple one field entry and DB can have only a few items. The items in your DB shall be encoded with full schema as it is defined by now. The purpose of vertical prototype is to early and quickly test basic software components and deployment infrastructure and frameworks as well as the key architecture patterns and thus to serve as a basic "scaffolding" for final product. It also serves as "teaching and training" tool to bring the rest of the team up to speed on development, frameworks etc. We recommend that back-end team be assigned the task of constructing this vertical prototype.

Link To Whoot Chat Login Page. http://whootchat.epizy.com/

2. Executive Summary

Current FAU students are given a number of promising tools to help them achieve their educational goals. FAU provides tutors, TA’s, help from the professor, and many other resources. If the school cannot accommodate every student's needs, there are third party options for help such as Khan academy, Chegg, etc. However, some of these options are one dimensional and only offer one form of help.

WhootChat is a website being developed as a way to help current and future FAU students trying to achieve higher education and certification. It will accommodate every students ever changing schedule, provide a social place where files, social tutoring, and knowledge can be shared. WhootChat will allow students to explore all aspects of the classes they will need to take in order to achieve the major they are seeking.

Since all students lead different lives and cannot use all the resources that FAU provides, WhootChat is here to help. The idea of WhootChat is to give current and future students a better understanding of the material and majors they are pursuing. WhootChat will do this by having two different features to the website.

The first feature will be a mapped major feature. Users will be able to select the major they wish to pursue and see all the past workloads for those majors. They will be able to review the ability of the professors to decide the best way the user can learn and if the professor is right for them.

The second feature will be a social chat session. You will be able to register and socialize with students from your current and past classes. Past students will voluntarily assist in any problems in the classes that students cannot solve with the resources available.

The most important thing WhootChat can do is help students achieve their educational goals. We believe that WhootChat can offer an excellent, safe, and educational learning environment while giving the students answers to any questions they may have.

3. Competitive analysis

.

Competitors features	Our Products features
- Chegg provides paid services for accessing notes and textbooks.	- Student provided notes
- Chegg is universal	- More localization
- Primary tutor and information interactment	- Discussion Board for Social interactions toward specified classes
- No suggested resources for students within a specific major, only links to tutoring and a general information link for	- Suggested Tools and Refreshers Page
- One dimensional use of content	- two dimensional use website for social and schooling

Competitive Analysis Summary

WhootChat will offer advantages such as free access to the website, social interaction, and local social assistance along with student provided notes. With WhootChat being free, the team feels that it would be more competitive than Chegg or Khan Academy. WhootChat would give the user the ability to ask previous students for assistance, in a particular topic/subject. It would also be a more organized and specific version of competitors such as Chegg. WhootChat would offer helpful tools and resources for the particular class you are currently taking.

4 . Data definition

- **HTML**: Hypertext Markup Language is the standard markup language for documents designed to be displayed in a web browser.
- **CSS**: Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML
- **Javascript**: programming language that conforms to the ECMAScript specification. It's high-level, often just-in-time compiled, and multi-paradigm.
- **PHP**: general-purpose scripting language especially suited to web development
- **Atom**: free and open-source text and source code editor for macOS, Linux, and Microsoft Windows with support for plug-ins written in JavaScript, and embedded Git Control, developed by GitHub.
- **Brackets**: source code editor with a primary focus on web development
- **OpenWeather**: online service that provides global weather data
- **Xampp**: free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages
- **Customer/User**: user searching for assistance on the website (students)
- **Class**: groups of students learning the same subject.
- **Subject**: area of knowledge being studied
- **Educational institution**: place where users attend class (for our project we will only be working with FAU students)
- **Administrator**: user responsible for the upkeep, configuration, and reliable operation of the page
- **Quick Links Tab:** section where useful links for students are displayed and can be interacted with
- **Chat Tab**: section where users can interact with each other and upload multimedia
- **View Profile Tab**: section where users can view and modify their profile

- **Tutors Tab**: section where users can find tutors available.
- **Majors Tab**: section where users can explore all aspects of their major through their correspondent major’s Hub.
- **Professors Tab:** section where students can explore different professors as well as read and write reviews.
- **Major**: area of study
- **TA**: teaching assistant
- **Professor**: academic rank at universities, post-secondary education and research institutions

5. Overview, scenarios and use case

WhootChat use:

- Will be localized for college students. In this case FAU. Every student will have the availability and resources each user is willing to upload.
- Main uses
  - Uploading and downloading documents such as examples of HW questions, and past students notes
  - Connecting socially with students who have taken previous classes
  - Connecting with students who are a part of your current class
- How the website would be used example
  - Before semester Starts User can review the class syllabus and connect with students who have already passed the class. The student can download old notes provided by students who willingly placed their notes for others to use and learn from and have access to any resources available before the class begins. When the semester and class starts the student can review previous quizzes/tests/homework to see what to study and which mistakes are common in class. While performing work the student can then ask for any "social assistance" from the users they have connected with. When the semester is over the and the user can willingly upload any HW/tests/quizzes that they believe can help the future students/users. They can then leave reviews of the class and recommend it to future students/users.

6. High-level functional requirements

1. Users should be able to create an account
    - Every user should be able to create an account by providing username and password. Once the account is created, the user should be able to login and log out.
2. User should be able to enter information on profile
    - Users should be able to enter their bio, major, year of graduation,and school they attend.
3. User should be able to upload multimedia
    - Users should be able to upload images, videos, files, and documents to the Chat Tab
4. User should be able to search any subject/major related items on the Search Bar
    - For example, users can search for notes provided by other students (willingly) on a specific subject.
    - Another example, students can search for usernames that correspond to students in their classes.
5. System should be able to provide "Main Page" section
    - Section where students can find information about their majors, professors, tutors, as well as modify their profile.
6. System should be able to to provide "Chat Tab" section
    - Section where students can interact with each other and upload multimedia like images, videos, files, and documents.
7. System should be able to provide a Search Bar
    - Students can type after clicking on the search bar and look for class notes as well as usernames.
8. System should provide a Security Management Module that helps reset passwords

7. List of non-functional requirements.

1. Storage Capacity
    a. Volume of data the system will persist at during run time
    b. Growth with user growth and usage
2. Security
    a. Login
        i. User or Admin
    b. Inactivity timeouts?
    c. Data back-up?
3. Audit/Maintenance
    a. File Characteristics - size before, size after, structure
    b. Time stamps for uploads and downloads as well as system maintenance.
4. Performance
    a. Response time, Browser refresh
    b. Processing times
5. Usability/Reliability
    a. Ability to perform its required function
    b. Response time/Processing times
6. Recoverability
    a. Back up frequency
7. Documentation
    a. User uploaded Documents
    b. Training Material
    c. Chat documentation

8. High-level system architecture and database organization

Lists of main software products, tools, languages and systems to be used, list of core APIs available at this point, supported browsers etc.

You also have to decide on which frameworks you will use if any. These provide both user interface, as well as cross-platform and cross browser layout/css. All external code you plan to use must be listed along with their license.

- Programming languages use:
  - HTML
    - Main page setup of web page design
  - CSS
    - Initial and creative adjustment of space created from HTML coding
  - Bootstrap
    - Creative adjustment coding
  - PHP
    - Functional programming for selected material that will allow for posting, searching, and links
  - Javascript
    - Function programming
- Editors used:
  - Atom
  - Brackets
  - Vsc
  - Node.js
- Xampp
- PhpMyAdmin
- Potential APIs used:

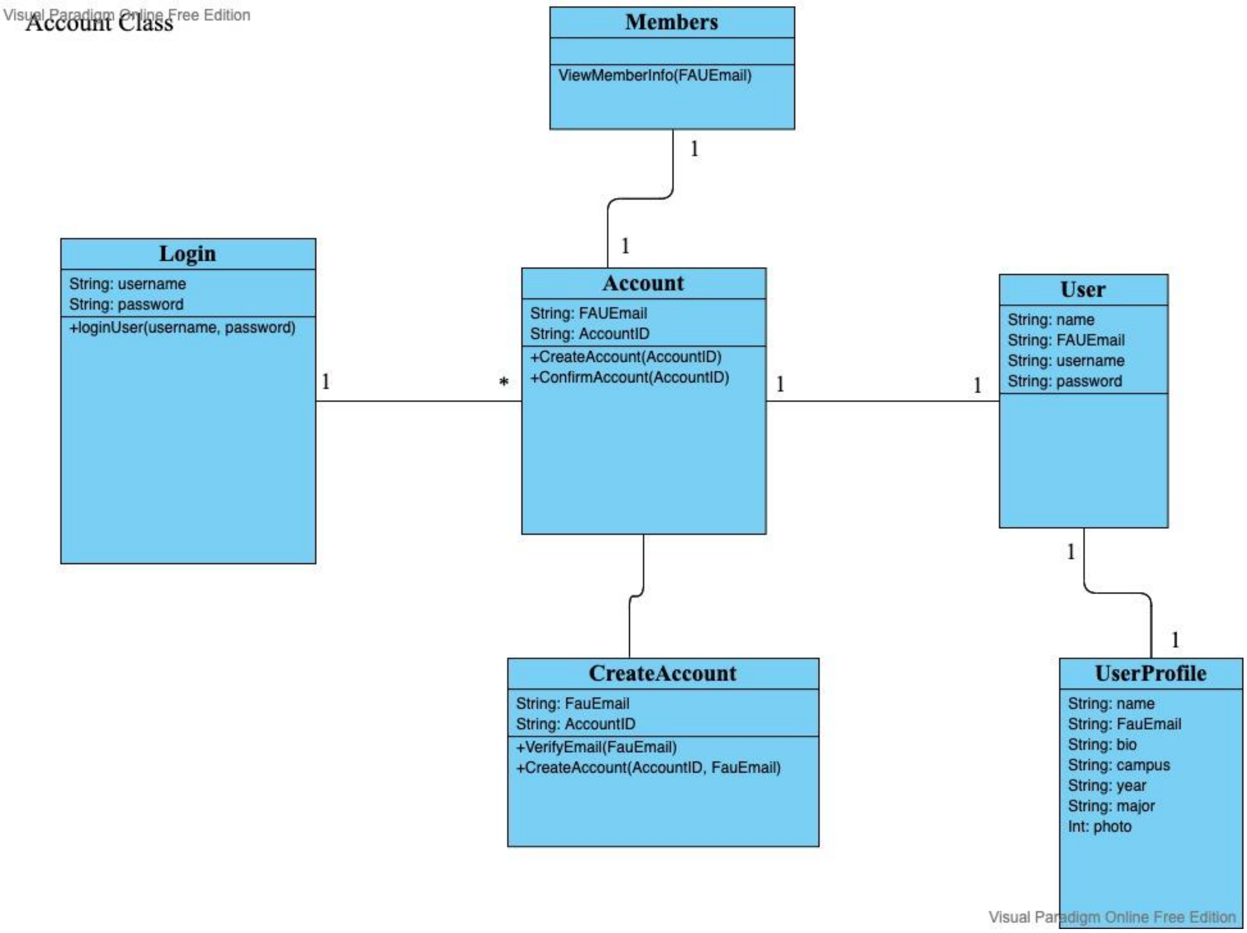
  - OpenWeather
- Supported Browser:
  - Google Chrome

9. High-Level UML diagrams

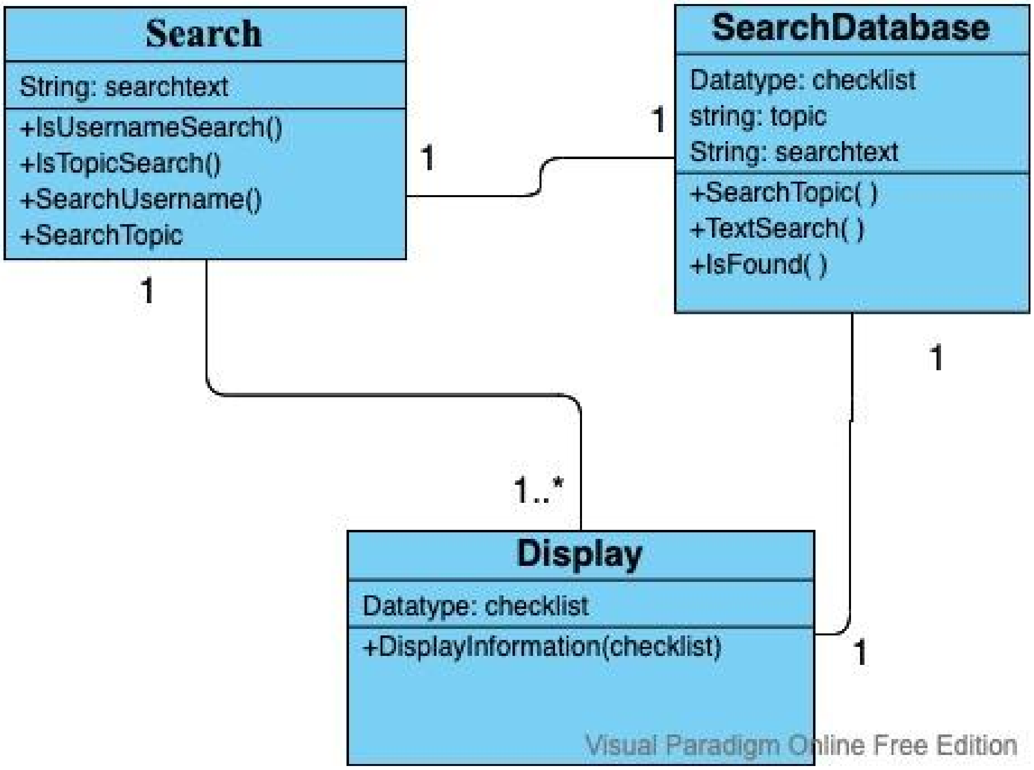
Familiarize yourself with Unified Modeling Language (UML). Find your favorite UML tutorials from the Internet. One good one is http://edn.embarcadero.com/article/31863

At minimum provide:

1) High-level UML class diagrams for implementation classes of core functionality, i.e. functionality with provided interfaces. Focus on main high-level classes only (one or at most two levels deep). This must reflect an OO approach to implementing your site.’

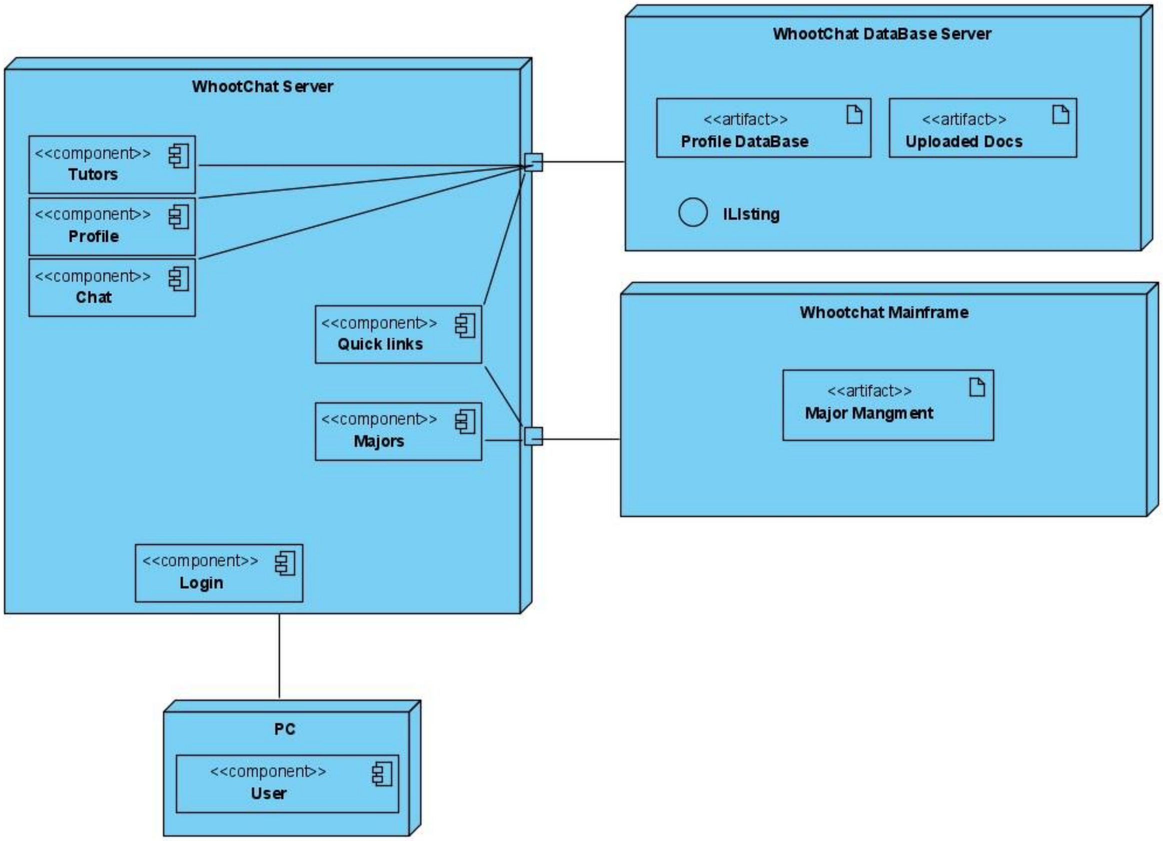


Search Bar Class



2) UML Component and deployment diagrams

Use data terms and names consistently with Glossary/Data Dictionary.



10. Identify actual key risks for your project at this time

Identify only actual and specific risks in your current work such as (list those that apply:

1) Skills risks:

Apprentice software developers. Possible Lack of knowledge when it comes to developing what we would like.

Resolve Risk: We can overcome this by studying, research, and building up the team's knowledge of software development. We can also gain guidance from other more experienced software developers. Another option would be to outsource this project to more experienced developers.

Lack of knowledge in software regulation, financial, and legal rigour.

Resolve Risk: Outsource to a legal/financial expert. Another option will be for some team members to research on their own to build the legal and financial expertise.

2) Schedule risks (can you make it given what you committed and the resources):

Server Maintenance and downtime on VM ware. May not have enough time to develop the software. Possibility of underestimating testing of software. Size of software could play a factor if it is too large.

Resolve Risk: Server maintenance and downtime can only be resolved by working ahead of schedule. Testing will need to take place as soon as possible along with implementation of vertical prototype. Another option could be creating our own free server.

3) Technical risks (any technical unknowns to solve), Getting a database to work on VM ware. Server Maintenance and downtime on VM ware which could affect our schedule. Possible power outage could erase certain data on VM ware.

Resolve Risk: testing prototype on VM ware and clearing up database issues with professor. Saving all work on Github and WinSCP. Creating our own server should resolve this issue.

4) Teamwork risks (any issues related to teamwork); Remotely working as a team seems to limit communication, teamwork, and problem solving.

Resolve Risk: communicating daily through emails, github, and group chats. Possibility of meeting as a group and working on site while adhering to COVID 19 protocols.

5) Legal/content risks (can you obtain content/SW you need legally with proper licensing, copyright). Possibility of users uploading books or copy written documents.

Resolve Risk: Admins will delete copy written work.

11. Submission

Store the modified Milestone 3 in your GitHub repo.

Each team submits one single word document with all the above required sections to Canvas by the due date. Must have a title page to your document.

12. Grading criteria

Your document needs to be well-written, well-organized (formatted) and reads well. Grading is based on cohesiveness and completeness.

1) Executive Summary	10 points
2) Competitive analysis	10 points
3) Data definition	10 points
4) Overview, scenarios and use cases	10 points
5) High-level functional requirements	10 points
6) List of non-functional requirements	10 points
7) High-level system architecture (UML)	10 points
8) Identify risk and actions	10 points
9) Working with GitHub	10 points
10) Vertical demo	10 points