

Midterm Project: First Demo Instructions

Overview

The purpose of the demo is to show progress and receive feedback. After the demo, you need to reconsider your project goals. Do you believe you will accomplish all the objectives you initially set by the end of the semester? Are you catching up on the schedule? Should you eliminate some functions or scaling down the project? Will you simplify or cancel some use cases? Or you think everything is going well and you can add more interesting functions to your current project?

Product Brochure

This contains a one-to-two-pages advertising leaflet providing basic descriptive information about your software product and distribute it to other students and the instructor. No more than two pages are needed. You should include the following information:

- Demo date
- Your name
- Project title
- Project webpage URL (on GitHub)
- The list of features and short description of each feature, system requirements to run the system, etc.
- A block diagram of the system and some screen snapshots could be helpful. You can think about it as a marketing brochure instead of a technical document.

Demo Format

The demo should focus on showing the operation of your system from the user's perspective. Minimize the spent time on non-essential features of your product, such as login, user authentication, and creating new user accounts, unless you implemented these in a novel, previously unknown approach. You should create some user accounts before the demo, so you can use existing accounts to demonstrate the system functionality.

Focus on showing the essential functionality of your system. Spending time on non-essential functions leaves the impression that there are no essential functions to show or disregard for time constraints (20 to 30 minutes per demo).

The demo format:

1. Up to 5 slides for a short overview of the product – what's the product, who it is used for, and the roadmap of your presentation
2. Live program demonstration that shows how the already-implemented use cases are working:

3. Briefly describe what each function/use case does, what business goal it helps to achieve
4. Demonstrate a scenario of how the users use it
5. During the demonstration, highlight the important aspects of your solution, such as elements of the user interface design, data processing at the application's back-end, etc.
6. Brief outline of your plan for the rest of the semester – what new features you are planning to have ready by the final demo.
7. You will deliver your presentation in a video like you did in the first week's introduction. You need to record your screen and your voice while displaying your product.

Step-by-step Guidelines for Screensharing and Recording your Presentation

Notes:

Focus on showing how your system solves an actual business problem. If your application uses a database, prepare several examples records before the demo and avoid creating new records during the demo. Use your allocated time to show your product rather than create another user profile or manage their accounts. Similarly, if your system offers different services to different user types (different initiating actors), create profiles for different user types before the demo. During the demo, show the services offered to these users.

You want to show that:

- You are addressing an important problem
- The problem presents technical challenges, which you successfully solved
- Your solution employs high-quality software design
- This system will work reliably, and be easy to understand and use

Because of time constraints, it is helpful to use a “roadmap slide” to guide the audience and make them aware of what is coming next.

Avoid showing source code, because there is no time to present the whole program and it is not possible to understand short segments code out of context and under time pressure. Instead of showing how the code works, explain what it does by using diagrams and animation.

When showing a table with data, or a chart, tell the audience what they are supposed to see in your data and charts – What are the key points they should see. All charts should show the labels and units on the axes.

Grading

Criteria	Points Possible
Coding	30
Unit Testing	10
Web design or Interface design	5
Interaction	10
Debugging	5
Document	10
Data collection	8
Database	5
Flyer	2
Slides	3
Project Management/ Project Plan	12

Demo Documents Submission

Using a file-sharing website share the folders below. If any item is not applicable to your project, state and explain in an extra explanation.pdf file.

The main folder should contain the following subfolders:

1. 1_code

Includes all your code with comments and README1.txt on how to run the code

2. 2_unit_testing

Program code to run unit tests and README2.txt on how to run the unit tests

3. 3_integration_testing

Program code to run unit tests and README3.txt on how to run the integration tests

4. 4_data_collection

Scripts or programs for data collection and README4.txt on how to run the data collection scripts

5. 5_documentation

Includes the following 5 separate documents with a cover page indicating what document it is, project title, your name. Include the table of contents where appropriate (PDF only):

1. System_requirements_documentation.pdf (assignment 2)
2. System_design_documentation.pdf (assignment 3)
3. Brochure/flyer.pdf
4. Presentation_slides.pdf (acceptable formats: .pdf or .ppt)

Upload your files on GitHub because sometimes the mail server removes code file attachments. Make them public, so that we should not need to login to access your files or search for them.