

LAB 7

Objectives of Lab 7

1. Team huddle to check on status of project design
2. Identify entities of a detailed design for your project
3. Create detailed designs for a subset of the entities of your project
4. Review your detailed designs
5. Assess your team's technical capability compared to the technical needs of the project
6. Team huddle to discuss next steps

Designers need to specify the details of the entities that make up the system. These definitions should be sufficiently detailed that the design can be given to a developer and the developer can create the entity as envisioned by the designer.

Once your team starts to develop a design, you should also be developing a better understanding of the technologies and skill levels needed to build the product. As a separate task, this lab will also provide a chance for you to assess your team's capability to work on the project and identify learning or skill development you may need.

Step 1: Opening Huddle

You know the drill!

Step 2: Draft a list of entities for your project

You should consider the following types of entities:

- Screens (or Web pages)
- Database tables
- Files (e.g., data that is stored as part of the system but not stored in a database)
- Code (modules, objects, or functions)

Use Figure 7-1 to list all the system entities that you can identify. A good way to start is to pick one area and focus on that. For example, if your system has a significant user interface, start by trying to name all the screens that would comprise your interface. For each entity you list:

- Enter a type, e.g., "screen"
- Give it a meaningful name, e.g., "CustomerProfile"
- Provide any short notes or explanation needed to identify the screen, e.g., "This screen captures customer information and preferences."

Step 3: Create detailed designs for at least 4 of your entities

You will not be able to design all the entities of your system in this lab, but this step will get you started. Pick 4 entities that you think you understand the best at this point and create a design for them. Every entity should have a name, type, and design details. Templates are provided to help you create detailed design for screens, database tables, and code functions.

Step 4: Review your detailed designs

After creating your designs, review them for completeness and clarity. Ask yourself this question: "If I was the developer and a designer handed me this design, would I know what to build without needing to ask a lot of questions?"

If you have created the design entities as a team, set them aside for a few minutes before reviewing each one. If you have worked in sub-groups within your team to create the designs, then exchange designs so the reviewer is a different person than the creator of a design.

Revise your designs based on the review.

Step 5: Assess your team's capability to complete this project

Once you have an architectural overview and the beginning of a design, you should be able to assess capability and identify things that someone on the team may need to learn. Use Figure 7-5 to summarize this information.

A – List the technologies you need for your project using the column on the left. Consider things such as programming languages, operating systems, specialized data sources, software libraries, support tools, and hardware.

B – List each team member at the top of a column, and then evaluate that person's knowledge of the technology in each row. For the column for each team member, use the following values:

- 1 – No knowledge or not much relative to the needs of this project
- 2 – Enough knowledge to accomplish part but not all this project
- 3 – Knowledge probably sufficient for this project

C – Discuss within your team how you will start to gain capabilities that you are missing. You do not need to turn in results of this discussion in this lab but will need to address this in the coming weeks.

Step 6: Team Huddle

Discuss status of all deliverables to date and plans for continuing work.

What to Turn In

In order to earn full credit for this lab, each team must submit:

1. Possible System Entities (see figure 7-1)
2. Detailed designs for at least 4 entities in your system. Use the templates in Figures 7-2 through 7-4 to get started
3. Team Capability Assessment (see figure 7-5)
4. Updated Project Gantt Chart

Rubric – 45 points

Item	Best	Average	Worst
Huddles (5 pts)	Quick, efficient, informative. All members are heard and valued.	Quick, efficient, and/or informative. All members are heard and valued.	Took too long, not efficient and/or all members not heard.
Entities List (5 pts)	5 or more good entities listed	3 to 4 entities listed and/or all choices are not reasonable	1 to 2 entities listed
Detailed Designs (20 pts)	4 or more entities designed with enough detail to implement them	Only 2 or 3 designs and/or designs are incomplete	Only one design or designs are poorly done
Capability Assessment (10 pts)	Technology and team fully covered, and assessment well done.	Mostly complete and assessments reasonable	Very incomplete or unrealistic
Updated Gantt Chart (5 pts)	Gantt chart shows accurate history of the prior phases and reasonable tasks for the design phase. Prototype phase is present but may not be fully defined.	Gantt chart shows history of the prior phases and reasonable tasks for the design phase. Prototype phase may be present but may not be well defined.	Gantt chart is incomplete or inaccurate in some substantive way. For example, no flow of tasks, no dependencies evident, no start and end dates.

Figure 7-1 – Possible System Entities

Product: Nimbus App

Team: 6

Date: 2/22/2019

Type	Name	Description or Notes
Screen	Home	Display basic info
Screen	Schedule	Show scheduled events
Screen	Add Event	Enter info to add new event
Module	DarkSky API	Fetch weather data
Module	Alarm Manager	Schedule alarms and execute code
Module	Notification Manager	Display notifications at times
Ringer	Notification Sound	Emit sound when notification is sent
File	Schedule	Keeps schedule information stored on device
Code	Weather Script	Develop easy way of displaying weather information (sent to notification manager)

Name: Add Event

Type: Screen

Purpose: This screen is needed to meet requirement 1.

Description: Figure 1 shows the layout for this screen. This screen allows users to add new events.

The screen contains the following elements: Start Time and End Time input boxes, checkboxes for days of the week.

Layout:

The diagram illustrates the layout of the 'Add Event' screen. It features a central vertical arrangement of elements within a rounded rectangular frame. At the top is a rectangular input box labeled 'Start Time'. Below it is another rectangular input box labeled 'End Time'. Underneath the 'End Time' box is a row of seven small, empty square checkboxes. Directly below these checkboxes are the letters 'S M T W R F S', each aligned with its corresponding checkbox. At the bottom of the arrangement is a rectangular button labeled 'Add Event'.

Figure 1 - New Event Screen

CI102

Name: Schedule

Type: Screen

Purpose: This screen is needed to meet requirement 2.

Description: Figure 2 shows the layout for this screen. This screen allows users to view scheduled events.

The screen contains the following elements: Event name, event time, event day for each event as well as an add event button.

Layout:

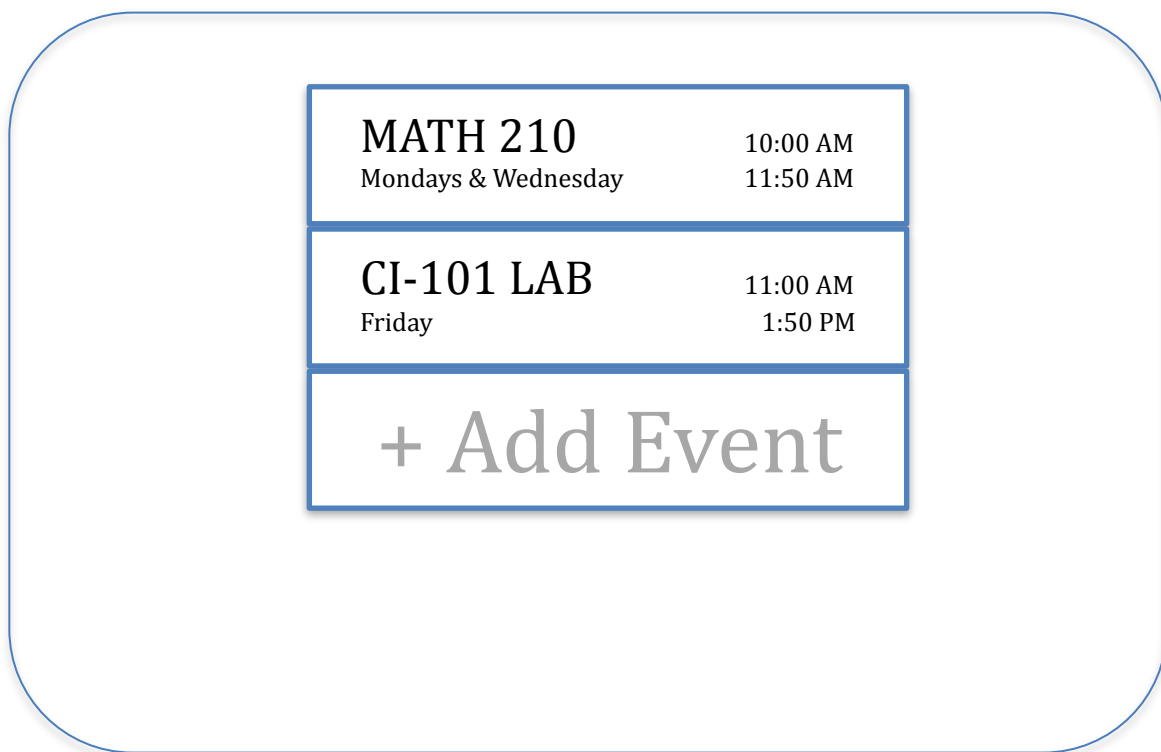


Figure 2 - Schedule Screen

Name: Weather Script

Type: Function

Purpose: This function is needed to meet requirement 3.

Parameters: The following parameters are used to call this function:

Name	Data Type	Notes
Start Time	Time	Start time of event
End Time	Time	End time of event

Return Type: String, information about the weather displayed in a nice way.

Processing: The function uses the start time and end time to call the weather API to gather information about the weather. The function processes this information and returns a well-worded and elegant sentence or two describing the weather.

Name: Alarm Manager

Type: Module

Purpose: This function is needed to meet requirement 4.

Parameters: The following parameters are used to call this function:

Name	Data Type	Notes
Alarm Time	Time	Start time of event
Repeating	Boolean	Whether this event is repeating
Warning	Integer	Time to notify before event

Processing: This module schedules an event at specific time (offset by warning time) that can be set to repeat each week (which our app will use for every effect). When the alarm scheduled goes off, it executes other code within our app.

Figure 7-5 – Team Capability Assessment

	David	Andrew	Joe	Hoff
Java	3	3	1	2
Android Studio	2	2	1	1
DarkSky API	2	1	1	1
Alarm Manager	2	1	1	1
UI Design	3	2	2	2
Data Processing	2	2	2	2
Marketing	1	1	3	1

** The table values represent an assessment of team member capabilities. The values are:

- 1 – No knowledge or not much relative to the needs of this project
- 2 – Enough knowledge to accomplish part but not all this project
- 3 – Knowledge probably sufficient for this project