SENG2021 Requirements and Design Workshop

Deliverable 1: Problem Statement/Features/Stories/Mockups

Deliverable Overview[1]

For this deliverable, students as groups need to identify a set of problems that they are going to solve in their project, devise the high-level requirements (we refer to them as features) and express these high-level requirements using a controlled language notation (refer to as user stories) where a user story captures features and scenarios.

Additionally, and in order to make the bridge between the written specification and a potential software system, the students need to develop a UI prototype to serve as a visual guide of defined features from the perspective of a typical actor. There are various degrees (known as “fidelities”) of prototyping. Students will be required to conduct both “low-FI” and “high-FI” UI prototyping. While in this phase you will likely be exposed to conducting somewhat a single iteration of the prototyping process, you may expect in real life situations, several iterations would be necessary until the system is fully understood in preparation for implementation. Having said this, we thus encourage you to conduct inter-group reviews of your prototype, to critic each other, and perhaps iterate over your prototype prior to submission for assessment.

Part1: Problem statement

Describe a few problem statements. Remember, while you are allowed to explore and negotiate with regard to the problems you are going to absolve, the following phases of the project will require you to abide by these requirements. Below is a summarised example from previous students, which should be used as guidance only.

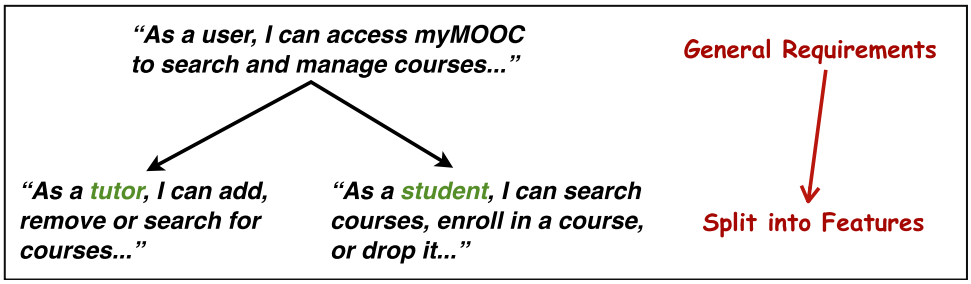
(1) There is no feedback system for a particular course.
(2) There is no reputation / popularity system for users or course instructors.
(3) The forums are visually messy. They are cluttered and jumbled up together.
(4) There is no general forum for the students to interact with each other, only a forum for each individual course.
(5) Old course content is removed / hidden, thus not allowing students to revisit already covered material.
(6) The dashboard is not interactive enough and lacks information of recent activity from the course.
(7) There is no Intuitive notification system.
(8) Major social hubs have been integrated poorly into the platform.

It is also possible to formulate a problem statement that targets a particular group of users or a community (e.g. teenagers, elderly or region specific).

1 The Deliverable guidelines are modelled according to the teaching material for SENG1031 created by Sci Prof Boualem Benatallah

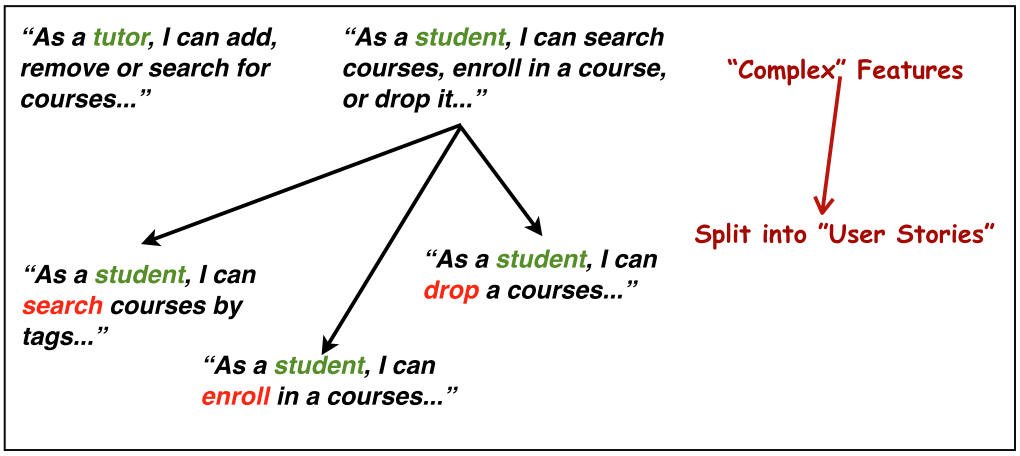
Part2: Identify and Describe User Stories

Identify high level requirements (i.e. features) based on the problem statement you have specified in Part 1 of the deliverable. The following is a summarised "working example" to be used a guidance in helping you prepare your submission:



The above starts with a general "requirement" (obtained from your problem statements. This is then decomposed into a "set of features" (where applicable). In this case, "people" are generic, which could be refined into two main categories of people, namely: Students and Tutors.

Subsequently, feature themselves may often be "complex", in which case they can be further split.



Notation for Describing a Feature:

Based on "Connextra Notation" (refer to Lecture notes for more details). Also recall SMART guidelines for effective presentation.

Note: *"effective presentation"* mainly means the presented user-stories should be (somewhat informally) evaluated by applying and verifying the SMART guidelines (at least with respect to: **S**pecific, **M**easurable, and **R**elevant). The other dimensions of **A**chievable and **T**ime-boxed are certainly nonetheless important, albeit will be discussed more succinctly during mentors' meetings, whereas the purpose of this deliverable is to focus your attention on the three dimensions mentioned above.

Finally, each finest-grained feature may then be expressed as “user-stories”. As mentioned, a user-story captures both: Description of the **Feature** + Description of a potential **Scenario**.

Notation for Describing a Scenario:

As a way for illustrating a single feature; often consists of 6-8 steps. Such as: GIVEN, WHEN, THEN, AND, … Note, this syntax can iterate, (e.g. GIVEN, WHEN, THEN, WHEN, THEN, etc.)

Putting it all together:

Feature *Search Tutor by reputation score*
As a *Student*

So that *I can find a qualified Tutor*

I want to *search Tutors by their reputation scores*

Scenario: *Search Tutors by reputation (cloud can be used to generate UI interactions/mock-ups…)*

GIVEN *I am on myMOOC home page*
WHEN *I click on “look for a Tutor”*
THEN *I should be on “Look for a Tutor” page*
WHEN *I fill in “reputation’’ higher than” “0.9”*
AND *I press “Search” button*
THEN *I should see all Tutors having reputation score higher than 0.9*

The following is also an **excerpt** from previous years’ students, in order to also be used as a **guide** in helping you prepare you submission:

Requirement: *Dashboard lacks a personalized, interactive calendar for dates related to enrolled courses*

Feature: *An interactive, personal calendar application on your dashboard*

As a: *Student*
So that: *I can easily view course related dates on my dashboard*
I want to: *View a personalized, interactive calendar*

GIVEN *that I am at my dashboard on Coursera*
THEN *I can see my personal Coursera calendar*

Feature: *(above) Calendar synced to course deadlines*

As a: *Student*
So that: *I know when assignments, tasks and tests are due*
I want to: *Sync my calendar with my courses to view these dates on the calendar*

GIVEN *that I am on my dashboard.*
WHEN *I look at my Coursera calendar.*
THEN *I can see dates related to my enrolled courses, such as assignment deadlines or course conclusion, for the current month*

Feature: *Add own dates to (above) calendar.*

As a: *Student*
So that: *I can plan my course work*
I want to: *Add my own dates to the calendar*

GIVEN *that I am at my dashboard and can see my Coursera calendar*
WHEN *I click on "edit" for the calendar*
THEN *I can include my own schedule onto the calendar*

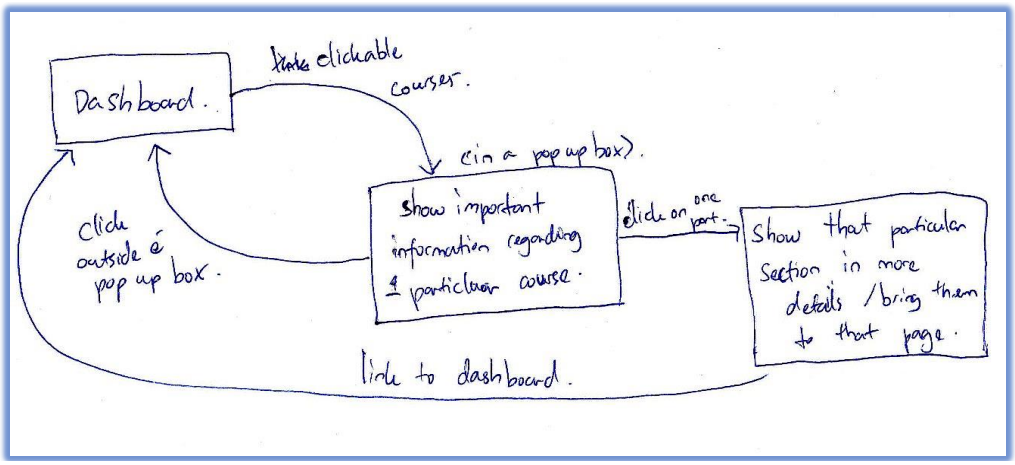
Part 3. Low-Fidelity Prototype.

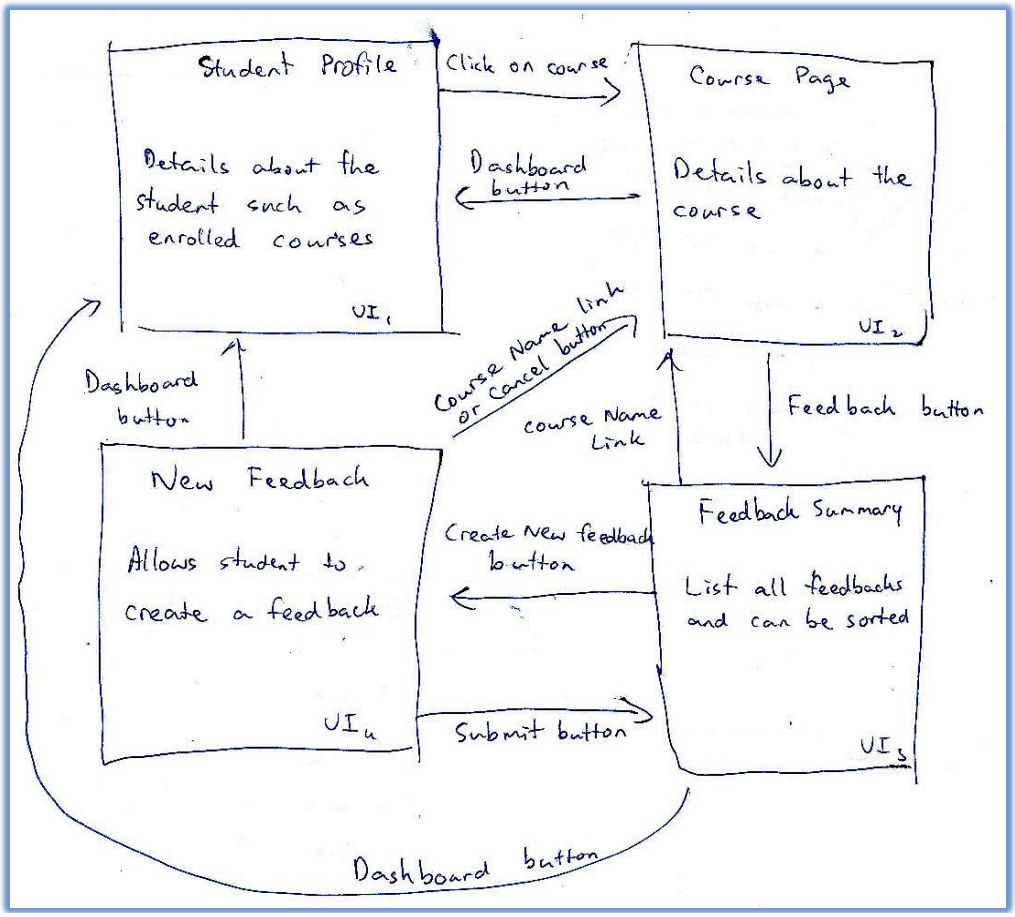
A Low-FI UI prototype often involves whiteboard and/or paper-pen sketching of the UI components. The purpose is to have an easy way to iterate over the design elements, without having to spend too much time and effort to redesign, when fast-paced changes are needed.

There are several aspects/dimensions in order to accomplish Lo-FI prototypes:

1. **Storyboard Interactions Graphs:** The concept of a storyboard has been borrowed from filmmakers, whereby it is usually utilised for illustratively describing the sequence of scenes (called a *'script'*) in order to focus on interactions not scene details. Likewise, storyboards in prototyping serves to illustratively describe the ***features*** of the system, while demonstrating the interactions between ***actors***. This helps present the transitions between UI components for others to understand, especially given a complex UI involving several UI components. A UI storyboard is often expressed using nodes and trees/graphs of UI screens (use cases involve sequences, decision points, alternative flows, etc.).

   To be used as a guide only, the following shows some examples from previous years' students work:



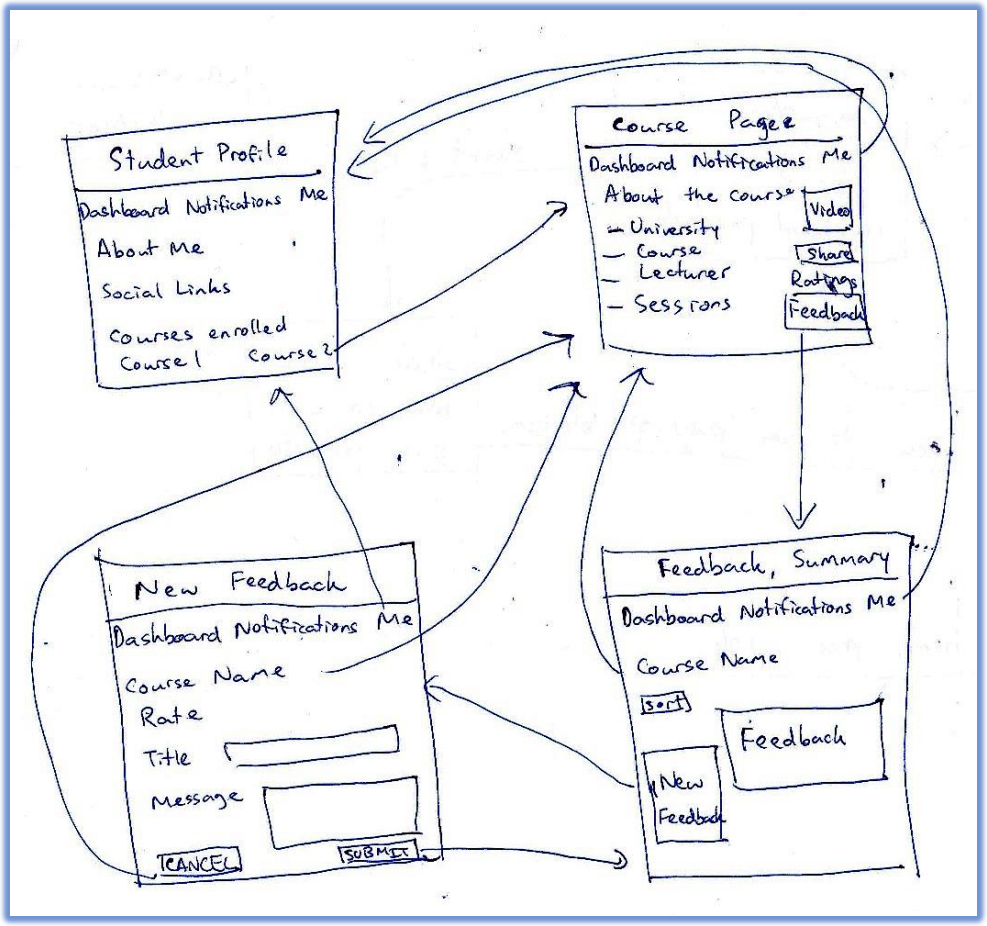


2. **UI Component Sketches:** As noted above, the first part of this deliverable will be defined and design the “big picture” storyboard, as shown above. This makes it easier to see and understand the overall interactions, without clouding with detail. The next step will be to define the detailed UI component sketches for each UI component. The purpose of this is to give the practitioner who is reading the prototype ***context*** about the specific UI elements that are involved in the interactions.

The UI component sketches referred to in this part, could either be presented as a stand-alone, (often in cases which explicit detail is required). However, it could also be combined in part (3) of this deliverable, as shown in the examples below!

3. **Storyboard Interactions + UI Component Sketches:** Therefore, in order to present the complete picture, showing both interactions between the components and the UI elements involved in the interactions… the storyboard is combined with the UI component sketches to create a more complete storyboard.

   Again, for your convenience, and to be used as a guide only, the following are some examples from previous years' students work:



Part 4. High-Fidelity Prototype.

In contrast, to low-Fidelity prototypes, high-FI prototypes use more advanced (often electronic) techniques for illustratively describing the prototype. Albeit, the methodology remains the same, with respect to: Storyboard interactions; UI Component sketches; and finally combing the above altogether.

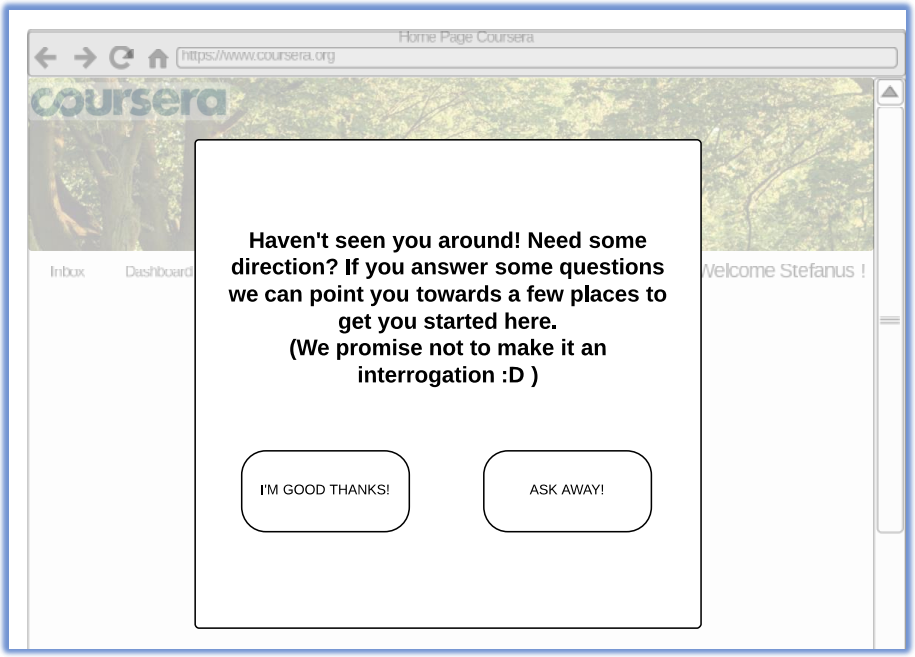
High-fidelity prototypes usually allow realistic (mouse-keyboard) user interactions. High-fidelity prototypes take you as close as possible to a true representation of the user

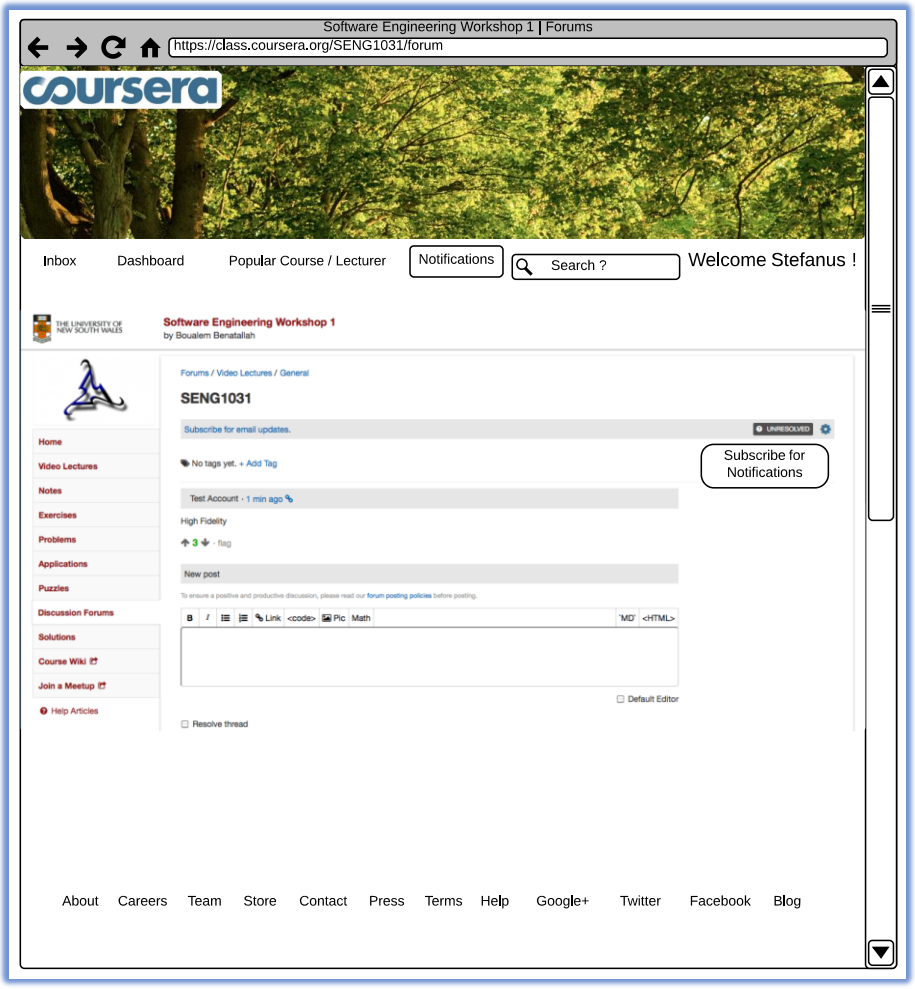
interface. High-fidelity prototypes are assumed to be much more effective in collecting true human performance data (e.g., time to complete a task), and in demonstrating actual products to clients, management, and others.[2]

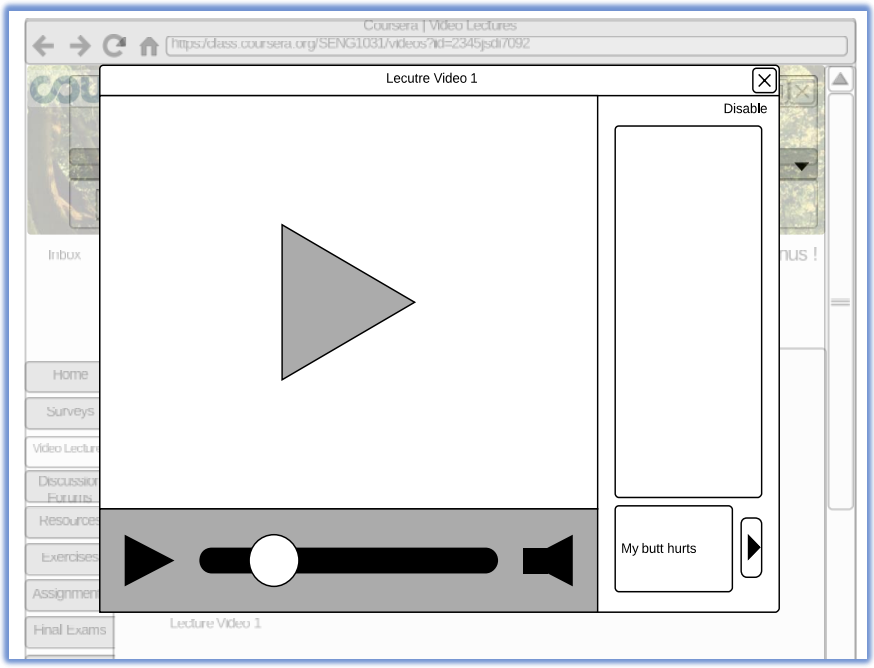
High-FI UI prototypes could be developed using wireframe tools (e.g., Axure, Luicid charts, Moqups…etc.) or using HTML/CSS/JS which require skills in code developments. We recommend for this deliverable to rely on HTML/CSS/JS for developing your High-FI UI prototype so that you can reuse parts of the code when building your functional prototype later on.

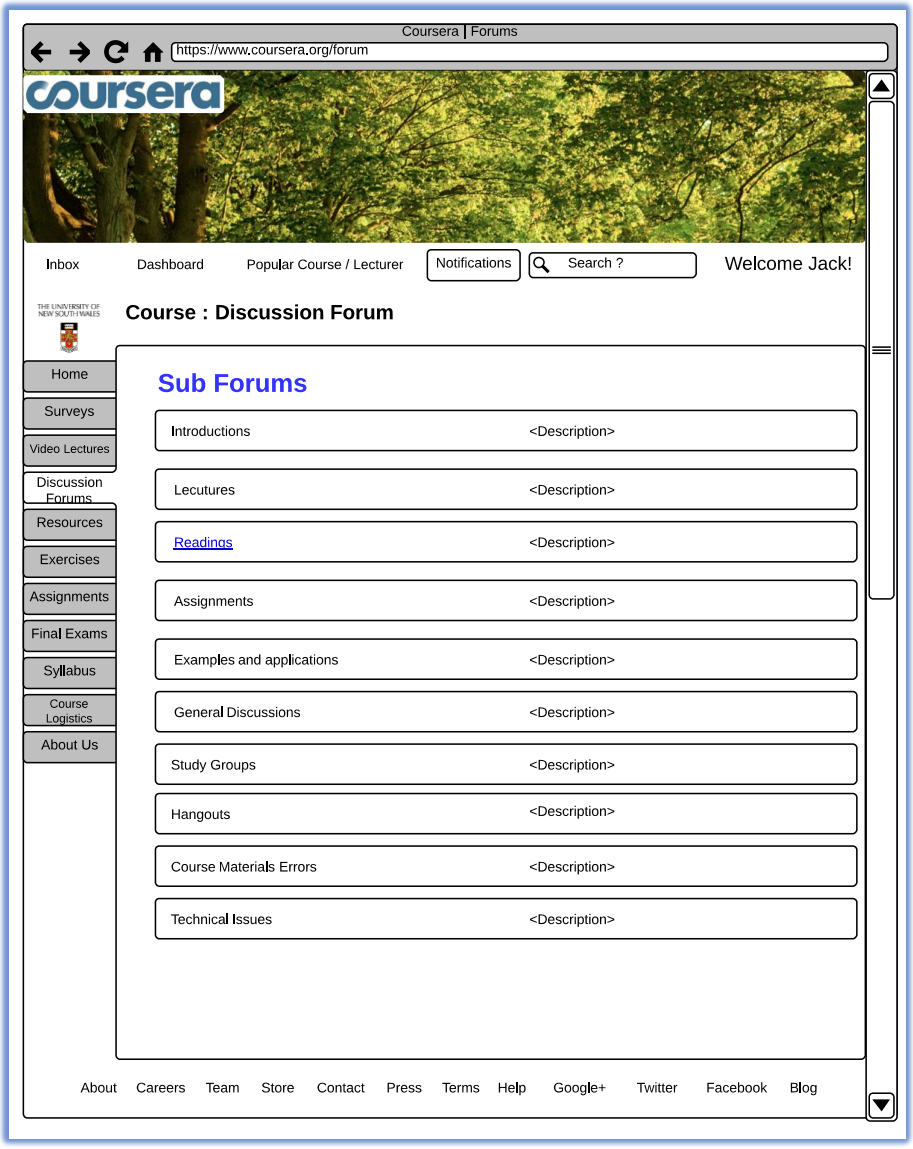
for purpose of example, the following are a small variety of sample screen layouts from previous years' students:

2 https://www.usability.gov/how-to-and-tools/methods/prototyping.html


Home Page Coursera
https://www.coursera.org
coursera
Inbox
Welcome Stefanus !
Haven't seen you around! Need some direction? If you answer some questions we can point you towards a few places to get you started here.
(We promise not to make it an interrogation :D)
I'M GOOD THANKS!
ASK AWAY!


Software Engineering Workshop 1 | Forums
https://class.coursera.org/SENG1031/forum
coursera
Inbox
Dashboard
Popular Course / Lecturer
Notifications
Search ?
Welcome Stefanus !
Software Engineering Workshop 1
by Boualem Benatallah
Forums / Video Lectures / General
SENG1031
Subscribe for email updates.
Subscribe for Notifications
Home
Video Lectures
Notes
Exercises
Problems
Applications
Puzzles
Discussion Forums
Solutions
Course Wiki
Join a Meetup
Help Articles
New post
Resolve thread
About
Careers
Team
Store
Contact
Press
Terms
Help
Google+
Twitter
Facebook
Blog


Coursera | Video Lectures
https://class.coursera.org/SENG1031/videos?id=2345jsdi7092
Lecutre Video 1
Disable
My butt hurts
Inbox
Home
Surveys
Video Lectures
Discussion Forums
Resources
Exercises
Assignments
Final Exams
Lecture Video 1


Coursera | Forums
https://www.coursera.org/forum
coursera
Inbox
Dashboard
Popular Course / Lecturer
Notifications
Search ?
Welcome Jack!
Course : Discussion Forum
Home
Surveys
Video Lectures
Discussion Forums
Resources
Exercises
Assignments
Final Exams
Syllabus
Course Logistics
About Us
Sub Forums
Introductions <Description>
Lecutures <Description>
Readings <Description>
Assignments <Description>
Examples and applications <Description>
General Discussions <Description>
Study Groups <Description>
Hangouts <Description>
Course Materials Errors <Description>
Technical Issues <Description>
About Careers Team Store Contact Press Terms Help Google+ Twitter Facebook Blog

Marking Scheme

PART	Excellent	Satisfactory	Poor
Part 1. Problem Statements *[10%]*	Well-thought Problem Statements are provided that correctly reflect the project topic. Selection of problem statements are concise (i.e. does not confuse irrelevant content).	Problems Statement are provided but is either: too generic; is too long or irrelevant; is not well-thought off, in that it would be difficult to translate these into potential features that could absolve these identified problems.	Problems Statements are incorrect and not well-thought. impossible to translate these into potential features that could absolve these identified problems.
Part 2. Identify and Describe User-Stories. ***(a) Features…*** *[15%]*	(a) Correctly examines an appropriate set of features that reflects the problem statement.	Features are proposed but partly does not correctly reflect the intention of the goals of the project, as identified in problem statement.	Features are not proposed, or incorrectly posed and does not reflect the intended goals of the project.
(b) User-Stories… *[25%]*	(b) Correctly describes the user-stories as: two main constituents: Description of the **Feature** (expressed using the correct notation) + Description of the **Scenario** (also expressed using appropriate notational language as presented in lecture notes). Concise responses provided. Stories presented according to SMART, with particular emphasis on: **S**pecific, **M**easurable, and **R**elevant	User-stories are lacking in accuracy as to how they are expressed. Syntax utilized is applied incorrectly. little evidence of taking into consideration the S, M R of SMART have been applied to propose the features	User-stories are not formulated, or largely incorrect, as well as incorrectly expressed. Hardly evident that any of the SMART methodology has been applied.
Part 3. ***(a) Lo-Fidelity Prototype.*** *[20%]*	All relevant features that were previously defined have been translated into aspects of the low-Fidelity prototypes. Such Lo-Fi prototypes should exhibit the various dimensions discussed above. Particularly Storyboard	The LoFI prototype represents some but does not encompass all the previously defined features. Or, it does not accurately reflect these defined features. In addition,	The LoFI prototype is significantly inaccurate, incomplete or illegible. It does not sufficiently reflect the proposed features; and/or does not

	Interactions Graphs and UI Component Sketches, as well as a combination of the above two aspects. It is expected that the interaction diagrams will be representative of a user-case, albeit showing the interactions of a typical actor in an illustrative manner.	the LoFi prototype is missing or inaccurately representing: the storyboard graphs to show the interaction between main actors; and/or missing or incomplete component sketches.	accurately contain the storyboard or component sketches
(b). High-Fidelity Prototype. ***[30%]***	All relevant UI screens have been successfully translated into a high-fidelity electronic prototype, with the help of using *a Mockups tool or HTML/CSS/JS*. Students should demonstrate sound understanding of designing a typical UI component, as well as demonstrate understanding how interactions can be registered with the help of *hyperlinks* and *hotspots*. Optionally, in some cases where "backend" prototyping is necessary (i.e. Apps/Web-Service interfaces), student should utilise tools such as Swagger (specs), in order to present their backend API.	HiFI prototype has been produced with reasonable quality. Or, it does not accurately reflect the LoFI prototype or the proposed features agreed upon. HiFI prototype may be broken or incorrectly implemented at certain places.	HiFI prototype is of significantly poor quality. It hardly reflects the proposed and agreed features, and/or does not reflect the LoFI prototype. Quality and depiction of UI components are ambiguous or would pose as misguided to a UI developer. The HiFI prototype hence does not serve its purpose.

Submission & Deadline:

You are required to prepare a document based on the above guidelines and submit it to through the online course management system. Submissions are required in PDF format. For the Lo-FI you can take snapshot of hand sketches on paper and bundle them in a PDF file. To share code repo or a link to Hi-FI prototype (please discuss with your Mentor the details)

The deadline for submission of the first deliverable is **Friday the 8th of March 2019 @11:59**. Please feel free to contact your mentors in order to discuss any further issues and/or details.