INFO30005 Project: Instructions

In this semester-long group project you will practice the skills and tools you learn in class by doing web design and development with a team.

Your task is to design and build a web application that meets the requirements provided to you in the separate Business Requirements document.

Figure 1: web dev team at work

Your Group

Projects will be conducted in groups of 5 students. All group members must be enrolled in the same weekly tutorial, and must attend the tutorial every week. (Absences must be justified in writing.) Groups will need to do substantial work outside tutorials as well.

Deliverables: overview

To help you make continual progress, and to provide multiple opportunities for feedback, the project is divided into three deliverables.

All of these are group (rather than individual) submissions. They sum to 50% of your whole-of-semester assessment.

#	Deliverable	Due	%
1	UI prototype + some HTML / CSS files	week 4	10
2	One complete feature: client + server + database	week 8	10
3	Complete web app + source code	week 11	30

Each deliverable is due by midnight Friday of the relevant week, Melbourne time.

Technologies and Tools

For equity reasons, you must use technologies taught in INFO30005 to build your web app.

- the web stack is: Node, Express, Handlebars, MongoDB
- supported by: authentication using Passport, database access using Mongoose
- the programming language is: JavaScript
- codebase is maintained on: GitHub
- web app is deployed to cloud platforms: Heroku, Atlas.

For prototyping we recommend and teach Adobe Xd, but will allow alternatives such as Figma on these conditions: alternatives will not be taught or supported, requirements for the deliverable and marking standards will not change.

Deliverable 1: details

Your first deliverable is:

- Design your web app's UI, and make a clickable prototype using e.g. Adobe Xd (7%)
- Implement two of the web pages in HTML and CSS (3%)

Make clickable prototypes of both the patient and clinician systems. Include all of the screens required to fulfill the business requirements. The patient prototype need only show phone-sized screens, while the clinician prototype need only show desktop-sized screens.

Implement in HTML/CSS the two static webpages "About diabetes" and "About this website". They should share the same external CSS file.

Submit via Canvas:

- a link (URL) to your Xd prototype
- a screenshot of your Xd project showing all artboards
- a link to your GitHub repo containing two HTML files and one CSS file.

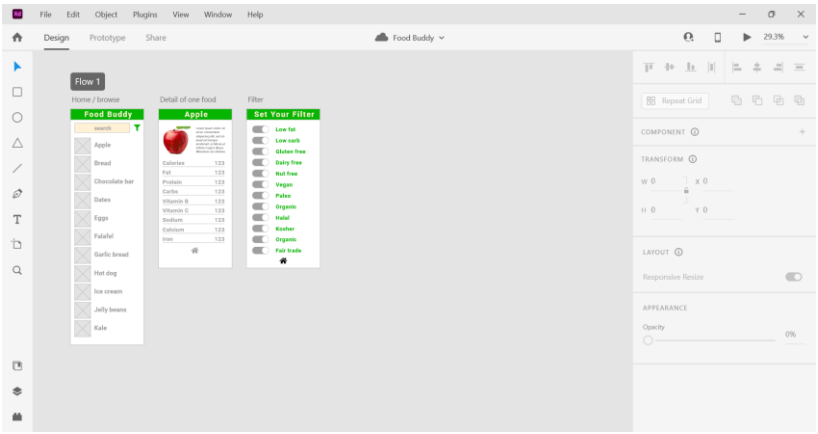

Figure 2: Example screenshot of artboards in an Xd project

Your work will be assessed according to these criteria:

- All business requirements are captured in the prototype
- Realistic data are shown on screens
- Interaction design and screen layout are of good quality
- HTML and CSS code is of good quality.

Deliverable 2: details

By the end of week 8, i.e. midnight on Friday 29th April, you should deliver one working end-to-end feature in your application, which enables a patient to enter some health data, and their clinician to see the data. For D2, your app should support the following scenario:

- Patient *Pat* has logged in and is viewing their home screen. Pat decides to enter today's blood glucose data, with an accompanying comment.
- This data is stored correctly in a MongoDB database on Atlas.
- Pat's screen changes to show that today's glucose data has been entered.
- Clinician *Chris* views their dashboard and sees the new data that Pat entered. The number is highlighted if it is outside Pat's safety threshold.

Note that many aspects of your app are not yet required in Deliverable 2:

- Logins (authentication via Passport) are not required in D2. Instead, you can simply hard-code Pat's and Chris's identities in your code.
- Validation of data entry is not required in D2. Your app can assume that Pat enters appropriate data.
- Software features described in the Requirements document but not listed on this page of the Instructions are not required in D2.
- You may manually place, in your database or screens, data other than what is entered in the scenario above; for example, Pat's safety thresholds, or other patient details for display in Chris's dashboard.

Your group should submit a single PDF that contains:

- Group name / group members / tutorial and tutor details
- The URL of your patient home screen, running on Heroku
- The URL of your clinician dashboard screen, running on Heroku
- The URL of your repo on GitHub.
- Tell us which commit in your repo you want to be marked. (It must be dated before the submission deadline. This is so you can keep building D3 while we mark D2.)
- MongoDB connection string that allows the marker to inspect your database on Atlas.

This is a group deliverable. It should be submitted by just one member of the group.

Your work will be assessed according to these criteria:

- Markers can use your app to successfully step through the scenario above.
- UI design is of high quality. The patient UI works well on phone and desktop screens.
- HTML, CSS and JS code is high-quality.
- The database design (structure of data in MongoDB) is high-quality.

See the submission page and rubric for further assessment details.

Deliverable 3 details

By the end of week 11, i.e. midnight on Friday 20th May, you should deliver the final version of your web application, with source code, offering the features described in the Project Requirements document.

Prepare your database for testing

Make sure your database contains at least the following data for testing purposes:

- At least 10 patients, including patient 'Pat'
- At least 2 clinicians, including clinician 'Chris'
- Pat is assigned to enter some but not all of the 4 possible time-series
- At least 2 patients including Pat have made at least 10 daily data entries
- Pat has > 80% engagement rate and is ranked in the top 5 patients
- At least 5 comments have been entered by patients

Submit one PDF that includes:

- Group name / group members / tutorial and tutor details
- URL of your patient login on Heroku
- URL of your clinician login on Heroku, if different to patient URL
- Usernames/passwords for patient Pat and clinician Chris
- URL of your repo on GitHub. State which commit should be marked.
- MongoDB connection string that allows the marker to inspect your database on Atlas.

The app will be tested using the following procedure:

Patient UI (10 marks)

Item	Description	Max marks
Log in	Entering incorrect details should result in an error message, while correct details should proceed to the patient app	1
Info pages	The two 'about' pages are visible, whether the patient is logged in or not.	1
Health data entry	Data entry form only allows entry of time-series that have been prescribed for this patient.	1
Health data entry	Patient can enter today's data, and can attach a comment to each number. (Markers will enter data that are both within and outside this patient's safety limits.) Data are stored appropriately, linked to patient, and timestamped.	1
Health data entry	Patient doesn't need to enter all today's data at the same time.	1
View data	Patient can view health data they have already entered, via an easily understandable layout, with dates/times visible.	1

View support messages	Patient can see the current support message from their clinician.	1
Engagement badge	Patient sees a badge if their engagement rate is over 80%.	1
Leader board	Patient can see a leaderboard showing the top five patients by engagement rate.	1
Change password	Patient can change their own password	1

Clinician UI (10 marks)

Item	Description	Max marks
Dashboard	Shows a table-like display with one row for each of at least 10 patients, one of whom is patient Pat.	1
Today's data	In patient Pat's row we can see the numbers that Pat entered today. Missing data, and numbers outside Pat's safety limits, are clearly marked.	1
Manage patients	Clicking on patient's name takes clinician to a page that allows them to see that patient's data.	1
Manage patients	Clinician can change which time-series a patient can record.	1
Manage patients	Clinician can change the safety limits for each of a patient's time-series.	1
Support messages	Clinician can change the current support message for patient.	1
View data	Clinician can view each patient's historical health data.	1
Clinical notes	Clinician can view existing clinical notes and enter a new one. Notes are tied to a specific patient, and timestamped.	1
View patient comments	Clinician has a page that shows all the recent patient comments. Clicking on a comment takes clinician to that patient's page.	1
Register patient	Clinician can register (add) a new patient to the system	1

Overall quality (10 marks)

Item	Description	Max marks
Security	App uses Passport for security and authentication.	1

Password safety	Passwords are encrypted in the database using bcrypt.	1
Input validation	Data entry is validated, both client- and server-side.	2
Database structure	Information about each user is stored as per Requirements.	1
Usability	Usability is good, navigation between screens is logical. Date/times are displayed in a user-friendly way.	2
Screen design	Screen design is of high quality. Patient UI is responsive.	2
Code	Code is of high quality.	1

Bonus features

Item	Description	Max marks
Charts	Numerical time-series can be viewed as charts.	1
Live alerts	Clinician UI shows an alert if any patient enters data outside safety range.	1
Custom colours	Patient can choose a colour-scheme for their UI. Preference is saved to database on a per-patient basis.	1

Bonus marks cannot increase the total mark for D3 beyond 30.

A note on calculating engagement rates

There is a potential problem regarding how to calculate a patient's engagement rate. It is made complicated by another feature in the app – clinicians can change a patient's data-entry requirements over time. This means that a patient's engagement rate on a given day depends on the history of clinician actions regarding that patient, which requires a complex calculation.

In response, teaching staff have decided to simplify the definition of engagement, as follows: *On a given day, a patient is considered to have engaged, if they entered any data that day.*

Thus for example, if a patient has been registered for 20 days, and entered some data on 16 of those days, their current engagement rate is 80%.

This will simplify what had become a complex calculation.

[end of Instructions document]