Opal Introduction

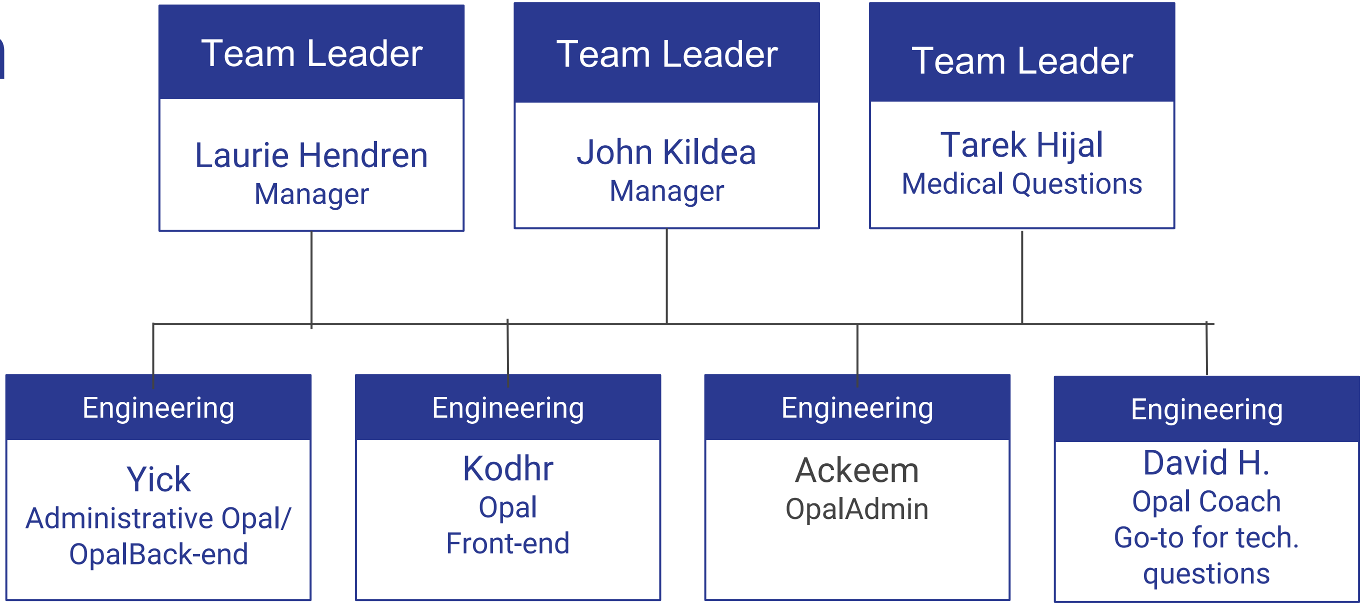
David Herrera

The HIG Group

- **Health Informatics Group,** Initiative started in 2014 by Laurie Hendren, John Kildea, and Tarek Hijal.
- **Goal**: Patient empowerment through information readily available and relevant to patients.

“ No one cares about you as much as you do”

The team



Team Leader

Laurie Hendren
Manager

Team Leader

John Kildea
Manager

Team Leader

Tarek Hijal
Medical Questions

Engineering

Yick
Administrative Opal/
OpalBack-end

Engineering

Kodhr
Opal
Front-end

Engineering

Ackeem
OpalAdmin

Engineering

David H.
Opal Coach
Go-to for tech.
questions

The Oncology Patient and Application

- Mobile application that serves as a hub between the patient and their health.
- Allows patients to stay informed and maintain a digitized version of their records readily available.

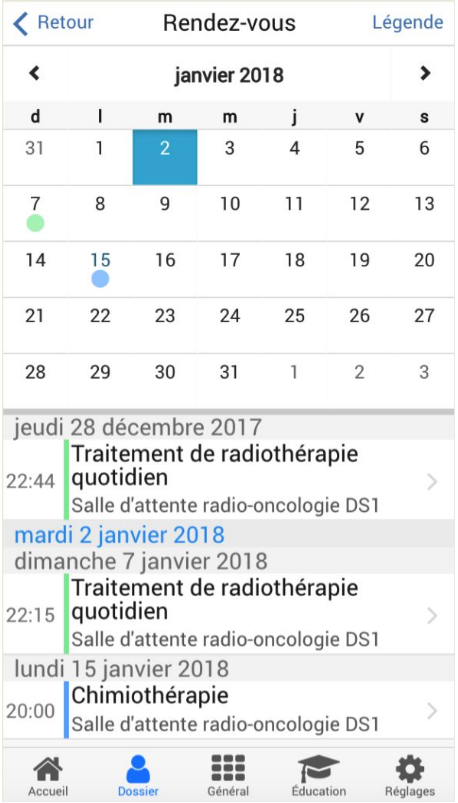
How does Opal do it?

- Offers patients personalized information tailored to their specific situation in a timely manner.
- Provides patients with their medical information and education on how to interpret it.

What does Opal offer?

- Announcements
- Appointments
- Checking in
- Clinical Documents

- Diagnoses
- Doctors
- Educational Material
- Notification
- Treatment Planning Status
- Treatment Team Messages
- Questionnaires
- … and more….



Rendez-vous

janvier 2018

d	l	m	m	j	v	s
31	1	2	3	4	5	6
7 ●	8	9	10	11	12	13
14	15 ●	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3

jeudi 28 décembre 2017

22:44 Traitement de radiothérapie quotidien
Salle d'attente radio-oncologie DS1

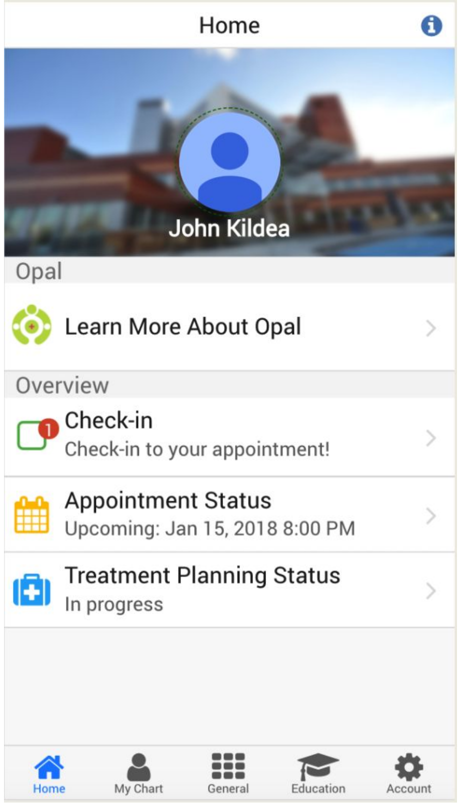
mardi 2 janvier 2018

dimanche 7 janvier 2018

22:15 Traitement de radiothérapie quotidien
Salle d'attente radio-oncologie DS1

lundi 15 janvier 2018

20:00 Chimiothérapie
Salle d'attente radio-oncologie DS1



Home

John Kildea

Opal

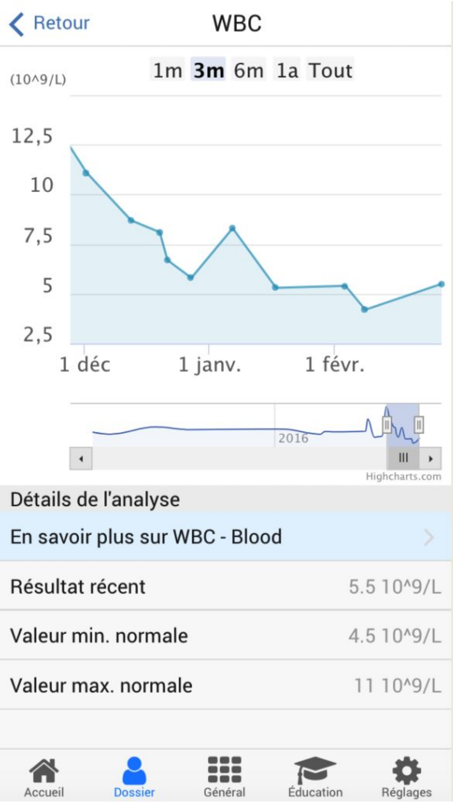
Learn More About Opal

Overview

Check-in
Check-in to your appointment!

Appointment Status
Upcoming: Jan 15, 2018 8:00 PM

Treatment Planning Status
In progress



WBC

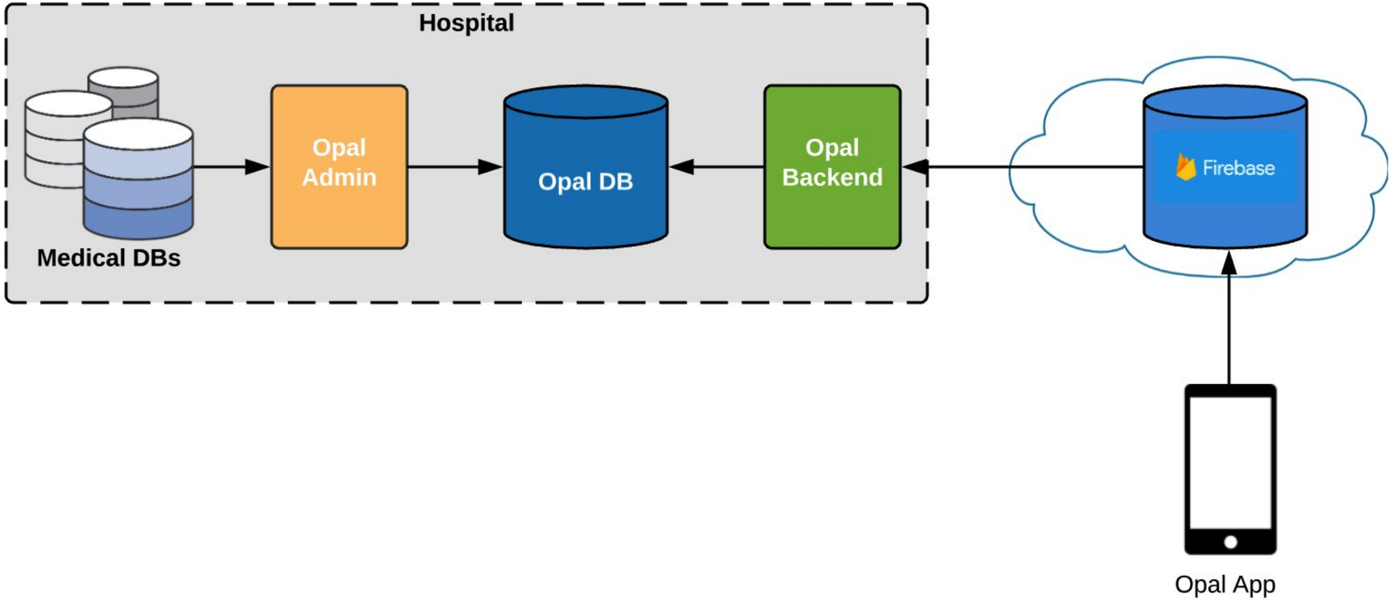
Détails de l'analyse

En savoir plus sur WBC - Blood

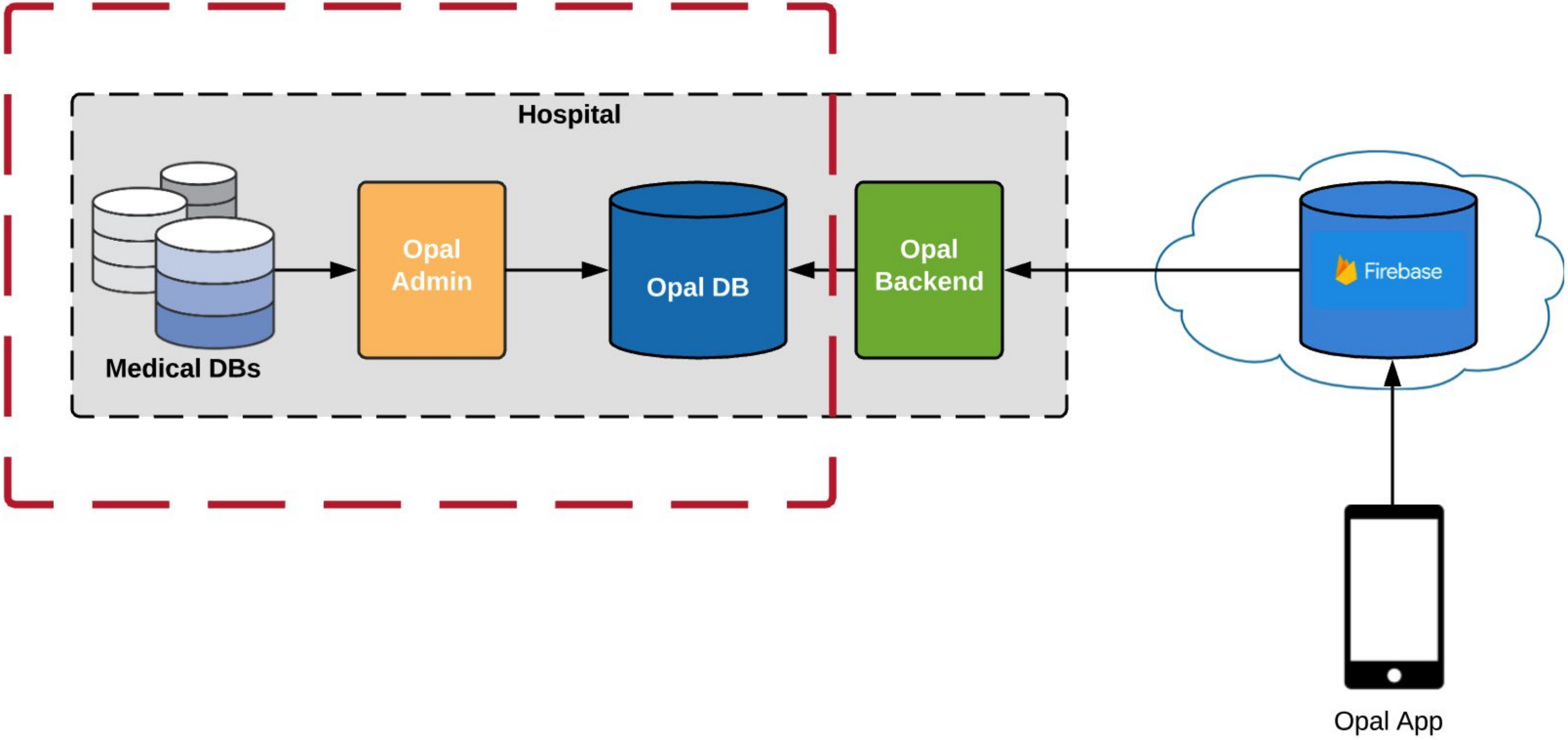
Résultat récent	5.5 10^9/L
Valeur min. normale	4.5 10^9/L
Valeur max. normale	11 10^9/L

Architecture (High-level)

Architecture high-level

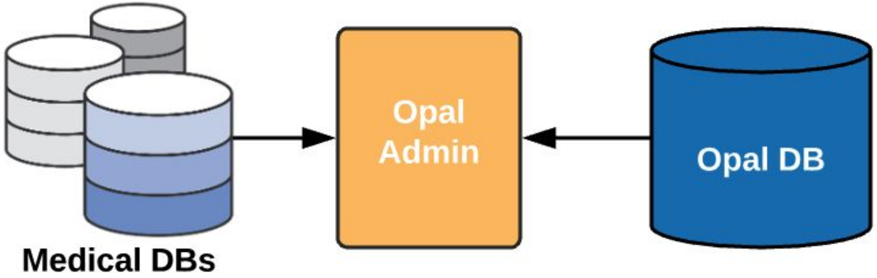


Architecture - OpalAdmin

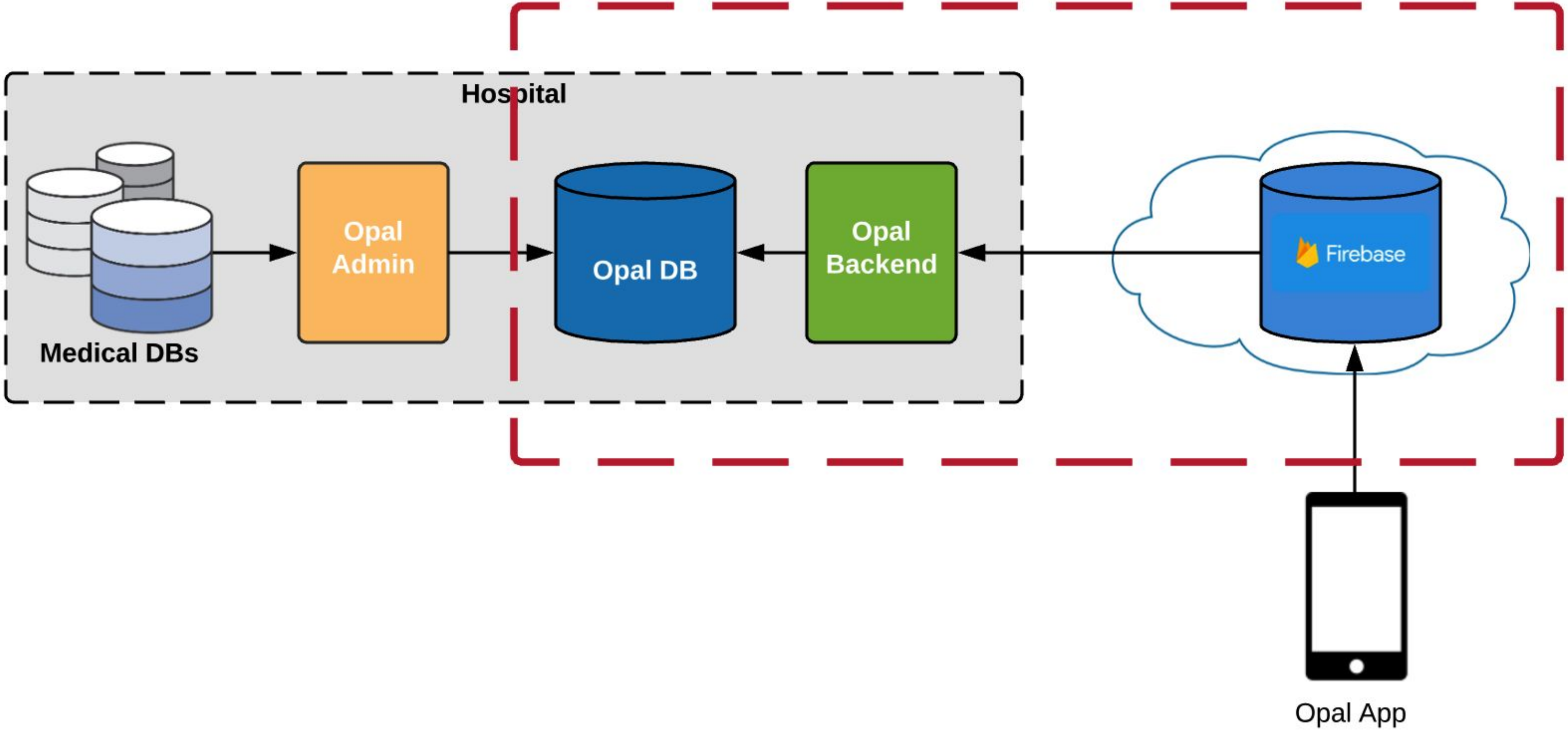


Architecture - OpalAdmin

- Queries the hospital DBs to obtain most up-to-date information
- Provides an interface to prepare personalized documents for the patients.
- Updates Opal DB periodically through a publishing interface
- Ackeem is the expert. Any questions regarding this should be directed to him

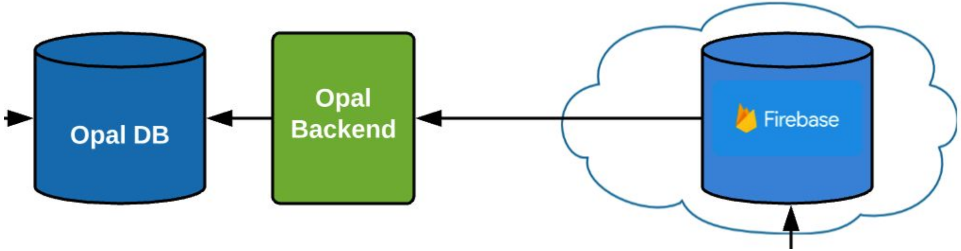


Architecture - OpalBackend

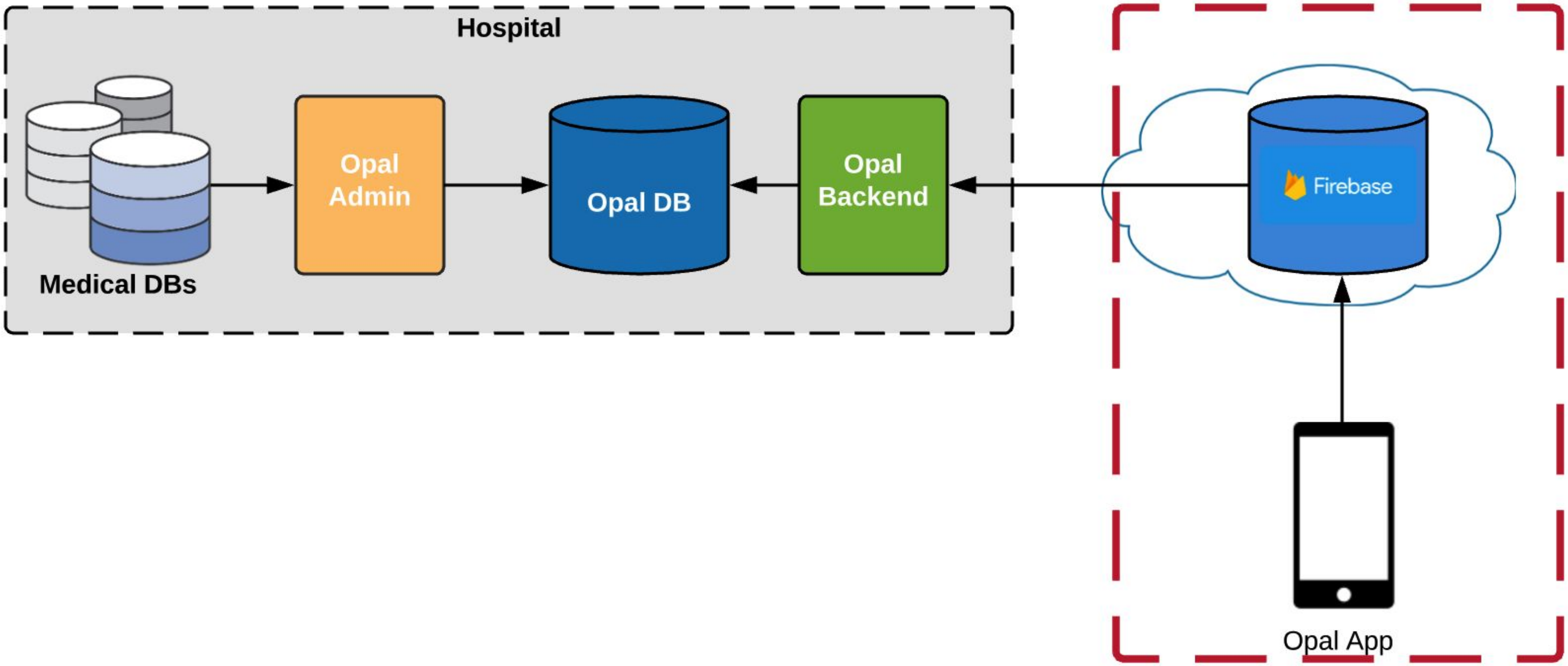


Architecture - OpalBackend

- Listens to Firebase for patient requests and serves as a back-end for the app.
- Queries OpalDB to update the patient.
- Updates OpalDB with patient-provided information from the app.
- Questions relating this can be directed to me

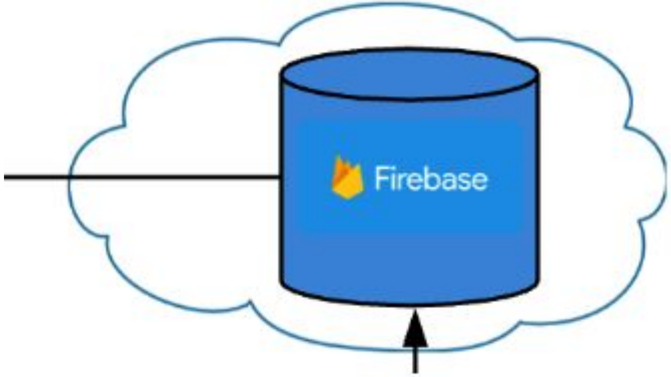


Architecture - Firebase and Opal App



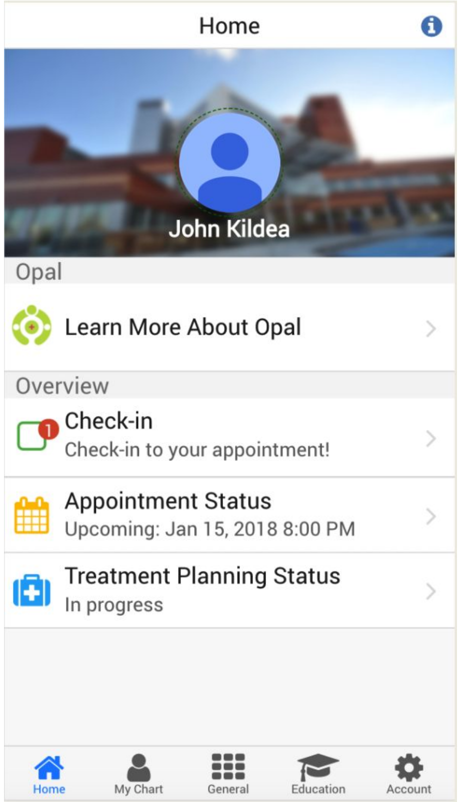
Architecture - Firebase

- Real-time Cloud Database owned by Google
- Use for patient authentication
- Provides a secure bridge between the hospital and the app.
- All data that goes into Firebase is encrypted
- Acts as an end-point for both the app and OpalBack-end



Architecture - Opal App

- Connects with the hospital through Firebase
- Sends request to Firebase and waits for response.
- Sends information to the hospital from patient. I.e. Feedback, questionnaire responses.
- Updates patient of any event such as a new appointment, clinical document, message etc.



Home

John Kildea

Opal

Learn More About Opal

Overview

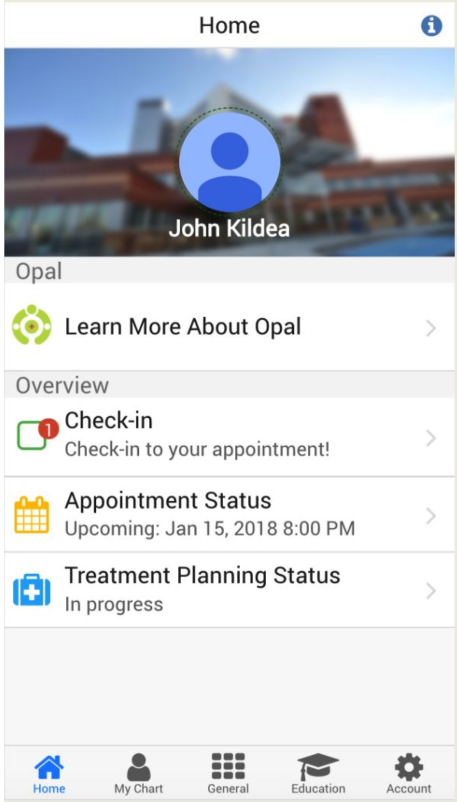
Check-in
Check-in to your appointment!

Appointment Status
Upcoming: Jan 15, 2018 8:00 PM

Treatment Planning Status
In progress

Technology Stack

Technology Stack - OpalFrontend



Home

John Kildea

Opal

Learn More About Opal

Overview

Check-in

Check-in to your appointment!

Appointment Status

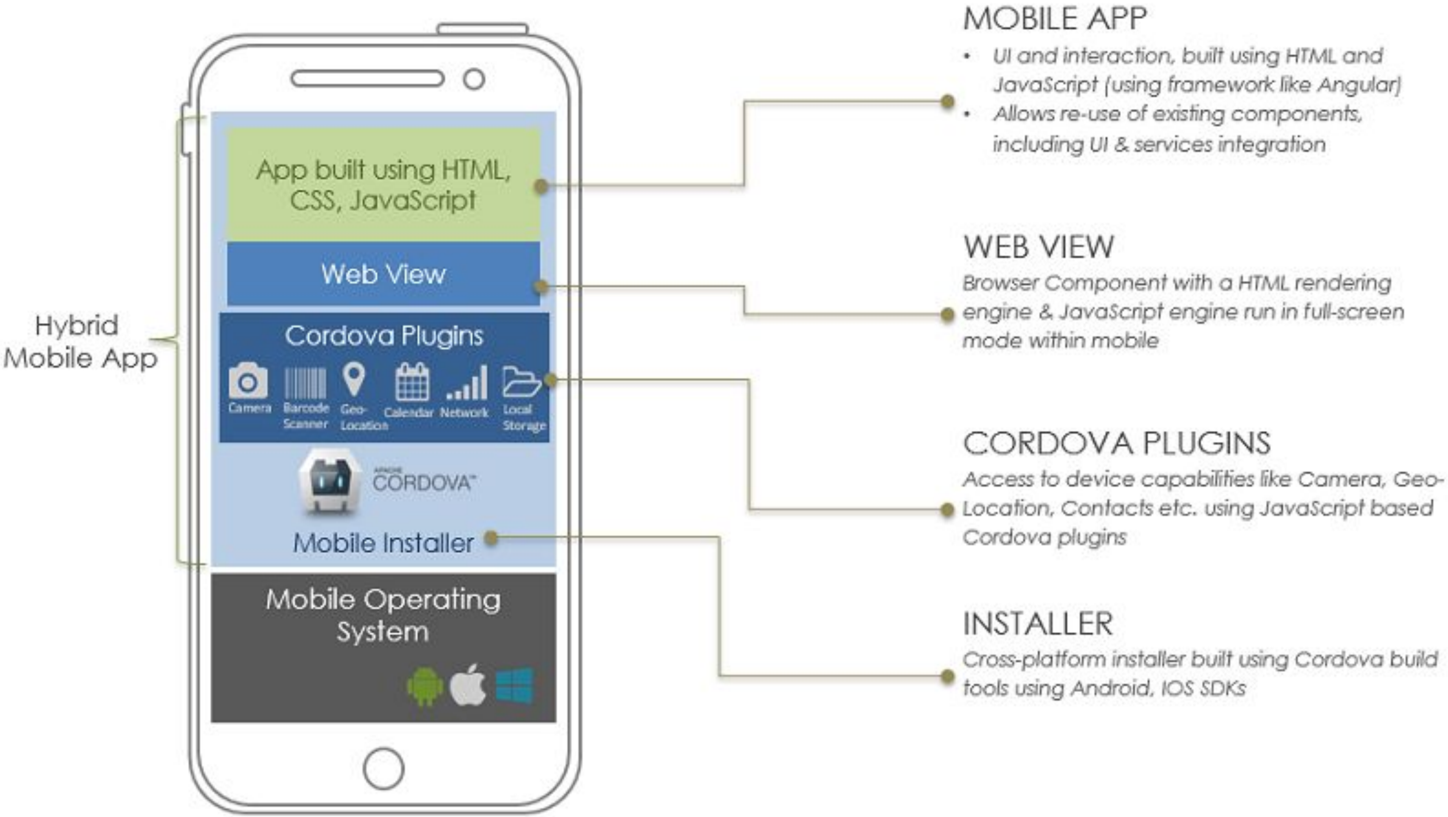
Upcoming: Jan 15, 2018 8:00 PM

Treatment Planning Status

In progress

APACHE CORDOVA™
Onsen UI
ANGULARJS by Google

Technology Stack - OpalFrontend - Cordova



MOBILE APP

- UI and interaction, built using HTML and JavaScript (using framework like Angular)
- Allows re-use of existing components, including UI & services integration

WEB VIEW

Browser Component with a HTML rendering engine & JavaScript engine run in full-screen mode within mobile

CORDOVA PLUGINS

Access to device capabilities like Camera, Geo-Location, Contacts etc. using JavaScript based Cordova plugins

INSTALLER

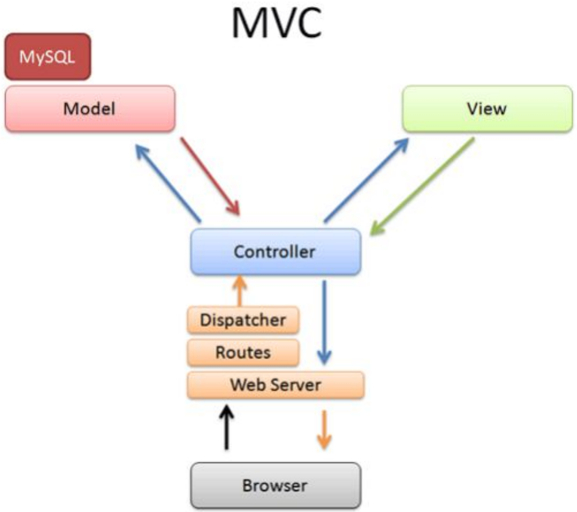
Cross-platform installer built using Cordova build tools using Android, IOS SDKs

Technology Stack - OpalFrontend - AngularJS

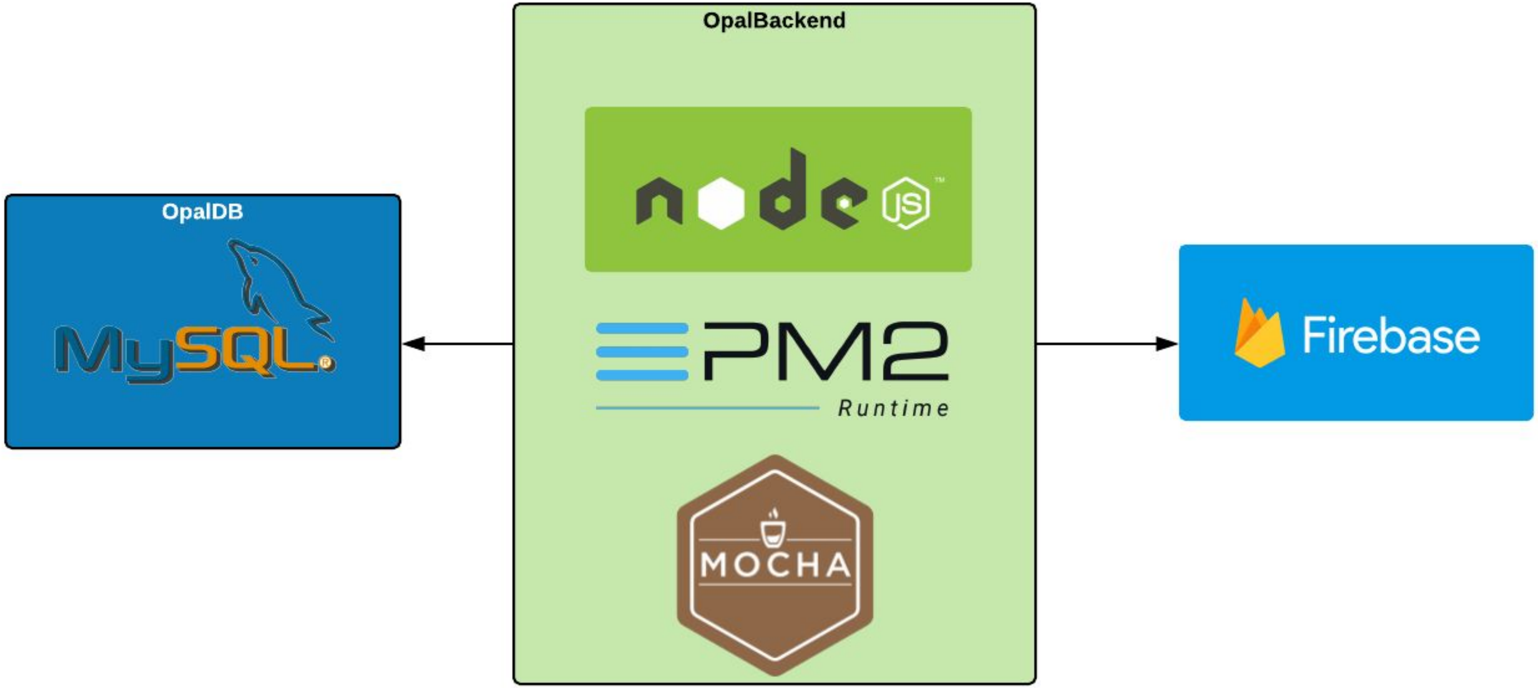
- **AngularJS,** JavaScript framework by Google.
- Revolutionized the way web apps were written via their two-way binding.
- Introduced one-page applications.
- Built on top of the **MVC** design pattern.
- Current version is v6, AngularJS still quite popular today.

Technology Stack - OpalFrontend - AngularJS

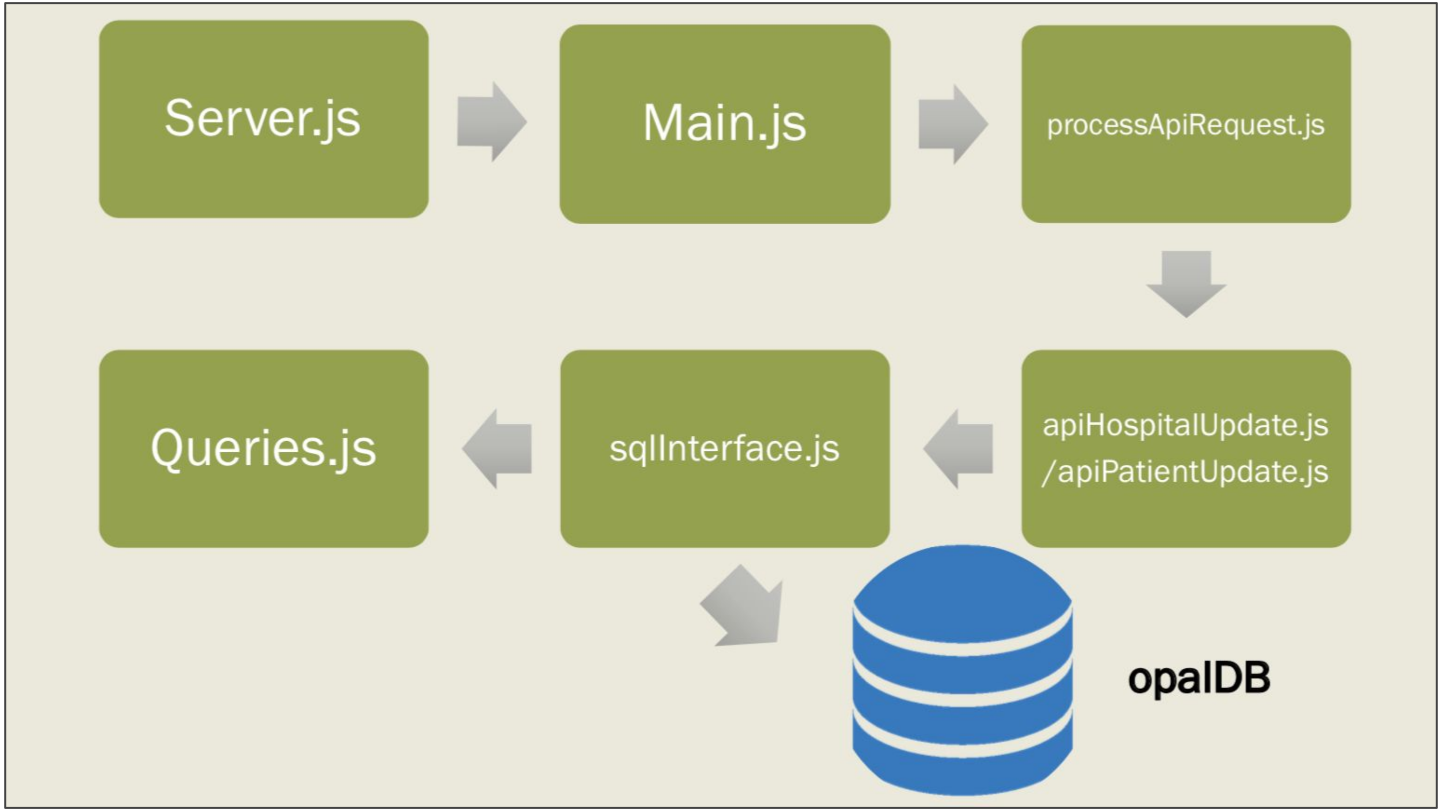
- **MVC:** Model, View, Controller
- **Model:**
  - Source of truth, maintains the state and the abstractions.
  - Queries the DB
- **Controller:**
  - Queries the DB either directly or through the model.
  - Provides the view with data and listens to the view to react to changes
- **View:**
  - UI, holds template of page. Interacts directly with the controller.



Technology Stack - OpalBackend



Technology Stack - OpalBackend



Week 1 - Boot Camp

Plan - Day 1 to Day 3

Day 1

Morning:

- Introduction to Opal
- Installation of Opal App and dev. env.
- Intro to the Opal Bug challenge

Afternoon:

- Reading resources, learning, reporting bugs.

Day 2

Morning:

- Front-end web development lecture
- Our front-end stack

Afternoon:

- Completing exercise for front-end

Day 3

Morning:

- Code review previous day
- Lecture on Opal’s middle-end

Afternoon:

- Completing exercise middle-end

Plan - Day 4 and 5

Day 4

Morning:

- Code review
- Opal backend technology stack lecture.
- Installation of local DB server

Afternoon:

- Adding backend “the opal way”, mini-project.

Day 5

Morning:

- Code review
- Lecture on Opal code infrastructure.
- Peer-review set-up

Afternoon:

- Putting it all together in the actual Opal app.

Challenges deep-dive

Challenge Day 2

Challenge Day 3

Challenge Day 4

Front-end

- Construct a mobile messaging app with AngularJS and OnsenUI
- Mock the backend
- Test and document your app components

Middle-end

- Use Firebase as a backend for your requests
- The app fetches from Firebase and immediately deletes this information

Back-end

- Use Node.js as the back-end for your request, listen to Firebase and update the Firebase upon request
- Delete any left-over info from Firebase.

Challenges deep-dive

Challenge Day 5

Opal - Integration

- Integrate your code in the Opal codebase

Week 2 - Projects

Plan - Day 1 to Day 3

Day 1

Morning:

- Lecture on MySql
- Individual Meetings for personal projects

Afternoon:

- Working on project

Day 2

Morning:

- Lecture on JavaScript, the good, the bad and the ugly
- Individual Meetings for personal projects

Afternoon:

- Working on project

Day 3

Morning:

- Lecture on Gulp, a task manager
- Individual Meetings for personal projects

Afternoon:

- Working on project

Plan - Day 4 to Day 5

Day 4

Morning:

- Lecture on asynchronous JavaScript

Afternoon:

- Working on project proposal and presentation

Day 5

- Presentations

Must have tools

- **IntelliJ WebStorm:**
  - Most complete IDE for Web Development
  - Testing framework integration
  - Database integration
  - Managing Tasks integration,
  - Very intelligent IDE with lots of help with code hints.
  - Free for students
- **GitKraken or Source Tree:**
  - Allows you to see visually the .git branch history and the repository status
  - Very important for code reviews to check your code before you push

Resources

- **AngularJS design guide**
  - https://github.com/johnpapa/angular-styleguide/blob/master/a1/README.md
  - Provides consistent and clean writing of AngularJS code.
  - Familiarize yourself!
- **CodeAcademy**
  - Intro. Tutorials to html/css/JavaScript/AngularJS/Node.js etc.
  - Can also use **Udemy, Udacity** (not free)
- **JavaScript**
  - You Don’t Know JS book series
  - JavaScript: the complete guide

End of Opal Intro

Opal Bug Challenge

1.
Install the mobile app

Khodr should have sent you an invite to the app.

2.
Install the Opal Development Environment

- Laurie should have given you an invite to join the Sable group.
- Go to https://github.com/Sable/qplus
- Follow the instructions there.

3.

Look for bugs!!!

Go through the different views in the app and identify bugs.

5.

Fix bugs

- After some bugs are reported, I will tag some of them as “good first bug”.
- Create a branch of opal_pre_prod
- Fix bug
- Create a pull-request for me to review.

4.

Report bugs

- Head to https://github.com/Sable/qplus.git
- Click on report issue
- Follow the format given

Opal Bug Challenge

- Points given based on:
    1. Quality and quantity of bugs reported
    2. Quality of Fix for a given bug.
    3. Extra points for testing.
- Winner is announced at the end of the week or beginning of next week. Prizes to be decided!

Questions?

- Wiki on Github
- If I am not present, email me at davidfherrerar@gmail.com
- Have fun and enjoy!