

# CS 837: Healthcare Application Development: Introduction

Term 2, 2023-24

T K Srikanth  
Jaya Sreevalsan Nair  
IIIT Bangalore



# Context of the course

E-Health Research Center (EHRC) at IIITB works on leveraging IT to improve healthcare services, with a focus on public health

Areas currently worked on: mental health, malnutrition, disabilities

Working with doctors, hospitals, govt to introduce IT solutions in various aspects of healthcare delivery.

Multiple apps and systems have been designed and some have been deployed

Course intended to provide exposure to such solutions, and understand the nuances and specifics of healthcare solutions

Digital health emerging as one of the major tech focus areas in the country. Ayushman Bharat Digital Mission and other digital initiatives will likely transform healthcare in the country, especially in less urbanized areas.

# Healthcare applications

Wide variety of IT-enabled solutions in healthcare

Our focus: systems that help manage patient data across clinics/hospitals

Typically involves data being collected & processed by multiple actors, across establishments and over long periods of time

# Course outcomes - application design

- Understanding the high-level architectures of web-based healthcare applications
- Understand designing of REST interfaces and microservices
- Translating user requirements to specifications and software design
- Understanding issues related to *usability*, *accessibility* and *responsive* design of (multi-lingual) mobile and browser-based healthcare applications
- Designing *security* features at the user interface, API, and data access levels, as well as security of data at rest and in transit
- Working with poor connectivity - *offline* mode

# Course outcomes - healthcare perspective

- Understanding design approaches to ensure ***privacy*** of health data - browser and mobile
- Understanding approaches to integration with external healthcare systems and interoperability of healthcare data
- Incorporating healthcare related standards into software designs
- Incorporating AI techniques to improve healthcare applications
- Incorporating Data Analysis and dashboards for healthcare

# Course project

One **end-to-end full-stack application** to be worked on over the semester

**Teams of 5 students each** - may change depending on strength of class.

Projects to be chosen from one of listed themes

**Final output:** Working prototype of an application and/or proof-of-concept of the use of one or more relevant technologies, along with relevant reports and documentation

**Initial requirements will be provided.** Detailed scope to be worked out in next 2 weeks

# Lectures/meetings

2 lectures per week (Fri)

Part of the lectures will also involve discussion about projects

Additional slots will be set up for project meetings

- Slots for each project theme
- to be finalized based on teams' availability
- one of the Fri slots may be used for project specific discussions

You are expected to attend lectures and the relevant project meeting slot(s)

# Teams

Form your own teams and update in shared sheet

- Team size should be as planned for the class

Teams should try to have coverage on front-end and back-end technologies - either existing skills/experience or willingness to learn quickly. UX skills would be useful in many projects

Try to self-assess team strength from standpoint of technologies, app development experience, and (object-oriented) programming skills

Each team member would be expected to have a good understanding of (and ideally contribute to) both the front-end and back-end of their projects, and how key features are implemented



# Project Selection

Create teams, add team members, choose project (can be changed) - by Mon Jan 8.

Each theme should ideally have roughly similar number of teams. Projects may be re-assigned in case of significant skew in the distribution of teams across project themes

# Assessment Plan

Each project will be planned with 4 milestones, roughly one a month. The progress in the project is assessed at each milestone.

- 45% of the grade is for progress at the interim milestones (15,15,15)
- 50% for the final demo and delivery (code, report, documentation)
- 5% for attendance - in lectures and in project discussion meetings (as recorded in CT)

Contribution of each team member will be assessed at each milestone. Grades may vary within a team

# Timelines

**Jan 12:** Project teams finalized, theme chosen

**Jan 19:** Initial requirements identified (through interactions in previous weeks)

**Jan 29:** Milestone 1: Presentation of high level functionality and key use cases and initial UI design draft

**Feb 23:** Milestone 2: Presentation of architecture, microservices, modularity, UI design, APIs; demonstration of basic features showing front-end to back-end integration

**Mar 29:** Milestone 3: Finalizing UI design, API's, db design; demonstration of key data flow, inter-app interactions (if relevant); core techniques (e.g. security, encryption) working

**Apr 26 (or exam week):** Final milestone: Demo of completed solution, code submission, project documentation

All presentations and deliverables to be submitted in LMS. Video of final demo also to be submitted

# Grade expectations

A	<b>Exceeds expectations.</b> All functionality/features working as planned, some additional accomplishments (e.g. UX/UI, design, implementation, incorporation of innovative technologies etc)
B	<b>Meets expectations.</b> Most use cases/features working, good UI, good software design and implementation, documentation
C	<b>Below expectations.</b> Significant functionality incomplete. UI and software design need improvement
D	<b>Does not meet expectations.</b>
F	...

Grade (above scale with +/-) assigned for the team's work

Individual contributions will be assessed and individual grades adjusted.

# What is covered and what is not covered

**Core requirement:** Applications involving full-stack development. Focus is on multi-role applications, involving collection/management/analysis of healthcare data - both clinical and process data, and with emphasis on usability, workflow, security, privacy, standards...

**Optional:** Use of open source frameworks/tools such as API gateways, HA, containerization/clusters, logging and log analysis, real-time dashboards, WAF, security audits (e.g OWASP). Incorporation of medical standards - FHIR, SNOMED CT, ICD10 etc. Accessibility. Compliance with GIGW.

**Add-ons:** Image/text analysis, ML/AI, Voice inputs, Analytics, sensors, AR/VR, Blockchain, Chatbots etc. However, can incorporate such features into projects - beyond the main full-stack development above

# Project Themes

1. Self-help therapy apps monitored by doctors - with activities, assessments, forums, chat
2. Tele-consulting platform (audio only - phone calls) with multiple clinics/docs
3. Tele-consulting platform (audio/video/chat)
4. Collaborative diagnostic platform - enabling clinicians/radiologists/pathologists to review and share comments on medical images
5. Hospital system integrated with ABDM sandbox, supporting common use-cases
6. App (phone/tablet) for field workers to extend health service from hospital to the home
7. Standards-compliant hospital app (SNOMED CT), and interchange of health data between hospitals (FHIR)
8. Hospital information system with tablet-based app for busy doctors

All projects will have two or more portals/apps, admin screens, some analytics/dashboards

# General guidelines

Use of “standard” or popular open source frameworks and components

- Server: Java with REST APIs. Springboot preferred.
- Client: Angular or React . Ionic/Capacitor for mobile apps if relevant
- DB: mySQL or similar systems

Focus on design of front end (flow, layout) and software modules

Incremental development of “vertical” functionality. Integrate functionality (UI and server) as you go

Constant focus on security and privacy - “privacy by design”.

Caution in using 3rd party plug-ins (security and ongoing support)

Generate sufficient test data to showcase performance and functionality

# Expectations: Basic features

Multiple portals/apps - for different roles

UX - with meaningful and relevant navigation and flow

Service/micro-service architecture with secure REST APIs

Data collection, storage and access - managing security and privacy

Modular, layered design

Ease of scaling/modifying functionality

Multi-lingual UI in all public-facing apps/pages

Notifications - SMS/email/WhatsApp or in-app

Basic dashboards



# Common considerations

- Management of patient demographic data - and ensuring privacy
- Management of patient health data - structuring across visits
- User on-boarding: self-registration or by admin/invitation - authenticating the user
- Login - and appropriate security mechanisms
- Security mechanisms, especially of API's, of transmitted data, and of stored data
- Anonymization of data for analytics/dashboard/reports
- Complying with privacy requirements - consent management
- Consistency of data/state across different apps (different roles/users)
- Admin/super-admin functionality
- Dashboards

For certain apps:

- Consent (where applicable)
- Integration with other healthcare systems
- Offline mode