

Queuing System

Daniel Istre Juan Trejo Bradley Shaw Mouhamadou Ndaw Harhsal Madhavapeddi
dri150030@utdallas.edu jct140230@utdallas.edu bjs170430@utdallas.edu msn160030@utdallas.edu hsm170030@utdalls.edu



CS 4485 / Fall 2020
Department of Computer Science
Erik Jonsson School of Engineering & Computer Science
The University of Texas at Dallas
Richardson, TX 75080, USA

ERIK JONSSON SCHOOL
OF ENGINEERING AND COMPUTER SCIENCE

Abstract

The Erik Jonsson School advising office has students check in by writing their name and other information on a sign-in sheet at the front desk before waiting to meet with an advisor. This information is then manually entered into a spreadsheet document for record-keeping. Our goal is to modernize and automate existing procedures, which rely on manual data entry, with an electronic system that offers a simplified self-service process for students through a mobile-friendly website and UTD single sign-on. Our solution uses the Django web framework and a MySQL database backend to provide a RESTful API for our ReactJS-based client application.

Keywords: API, Full Stack, UI/UX

Results

UTD ECS Check-in Reports History Public Queue

Refresh

	Name	UTD ID	Major	Wait Time
Check In	Galen Denison	8633278192	Computer Science	12 min.
Check In	Maison Juarez	7043450229	Computer Engineering	8 min.

Name: myj331134

UTD ID	TODO [Orion]
Major	Computer Engineering
Reason	75 Hour Audit
Reason note	
Degree Plan Year	TODO [Orion]
Advisor	aes111111
Advisor Note	

	Name	UTD ID	Major	Wait Time
Check In	Carmen Clarke	5344036896	Biomedical Engineering	11 min.
Check In	Kali Kent	1416080843	Software Engineering	10 min.
Check In	Javan Lindsay	5738102696	Electrical Engineering	4 min.

(Above) Advisor selects waiting visitor

(Below) Visitor receives notification

UTD ECS Check-in

Your advisor is ready to see you. X

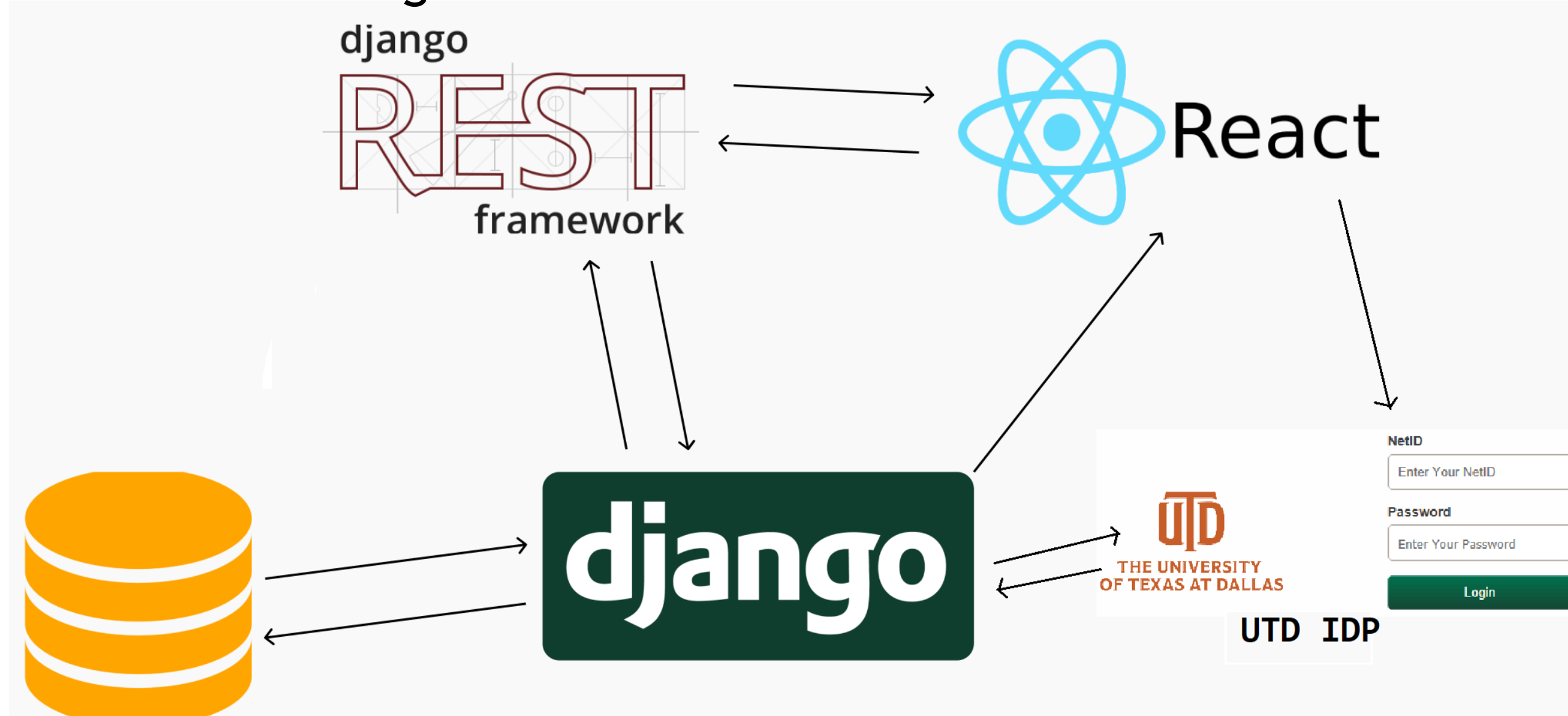
Meeting status:

Advisor	Location	Ready
Amy Sharp	ECS 2.259	Yes

Cancel

Architecture

- Django, Django REST Framework, and MySQL backend
- ReactJS frontend
- UTD SSO integration for user authentication



Impact

- Self-service and notification for advising visitors.
- Reduces administrative load by automating queuing logic, providing visitor information to advisors, and generating reports from visit records.

Performance

- 8 out of 12 minimum viable product deliverables achieved.
- Critical integration with UTD single-sign-on, providing authentication, not implemented.

Summary

- Students and other visitors can check in to meet with an advisor through a ReactJS web application on a mobile or desktop browser. Visitor receives notification when selected.
- Advisors can view visitors waiting to be seen and select one from the queue. Advisors can set their availability status.
- Reports can be generated with various filter parameters such as date ranges, advisor, NetID of student, etc.