

MICHIGAN STATE

U N I V E R S I T Y

Project Plan

Training Scheduling and Optimization System

The Capstone Experience

Team United Airlines

Matthew Libiran

Brian Lowen

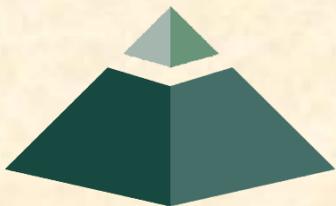
Nathan Rizik

Kailash Saravanan

Hydra Xu

Department of Computer Science and Engineering
Michigan State University

Spring 2019



*From Students...
...to Professionals*

Functional Specifications

- Redesign the client's 20- year-old maintenance training course scheduling system
- Current system has scattered functionality and unintuitive interface
- Maintain previous functionality while adding improvements
- Implement optimizer that suggests the best schedule based on a variety of attributes



Design Specifications

- Scheduling of classes
 - View and manage courses
 - Assign multiple instructors to a single course
 - Automate location suggestion based on resource availability
- Instructor time tracking
 - Tracks time with exception codes through a calendar and table view
- Course request ability
 - Allows a scheduler to be notified whenever a guest requests a new course to be scheduled
- Rosters and attendance tracking
 - Keep track of attendance within each course
 - Export a class completion roster on the final day of the course
- Schedule Optimizer
 - Suggest an optimal schedule based on classroom resources and instructor availability



Screen Mockup: Schedule

The screenshot displays a web application interface for a Learning Scheduling System. The top navigation bar includes the United logo, the system name "Learning Scheduling System", and the user name "Kailash Saravanan". A left sidebar contains navigation links: HOME, CALENDAR, EXCEPTIONS, SCHEDULE (highlighted), COURSE CATALOG, and WHO CAN TEACH. The main content area features a "Table" and "Calendar" view selector, with "Table" selected. To the right are "Generate Schedule" and "Add Class" buttons. Below these are several filter dropdowns: From Date, To Date, Course Code, Catalog, Location, Instructor, Point, and Cancelled. The central "Schedule of Classes" table lists ten identical entries for course A456789, titled "B757 Systems Avionics (2 week)", scheduled from 1/20/19 to 1/27/19 at 07:00 in room 001D87, located in DEN, with instructor Musk, Elon.

Schedule of Classes										
Action	Course Code	Course Title	Start Date	End Date	Start Time	Room	Location	Point	Instructor	
	A456789	B757 Systems Avionics (2 week)	1/20/19	1/27/19	07:00	001D87	DEN	OPC	Musk, Elon	
	A456789	B757 Systems Avionics (2 week)	1/20/19	1/27/19	07:00	001D87	DEN	OPC	Musk, Elon	
	A456789	B757 Systems Avionics (2 week)	1/20/19	1/27/19	07:00	001D87	DEN	OPC	Musk, Elon	
	A456789	B757 Systems Avionics (2 week)	1/20/19	1/27/19	07:00	001D87	DEN	OPC	Musk, Elon	
	A456789	B757 Systems Avionics (2 week)	1/20/19	1/27/19	07:00	001D87	DEN	OPC	Musk, Elon	
	A456789	B757 Systems Avionics (2 week)	1/20/19	1/27/19	07:00	001D87	DEN	OPC	Musk, Elon	
	A456789	B757 Systems Avionics (2 week)	1/20/19	1/27/19	07:00	001D87	DEN	OPC	Musk, Elon	
	A456789	B757 Systems Avionics (2 week)	1/20/19	1/27/19	07:00	001D87	DEN	OPC	Musk, Elon	
	A456789	B757 Systems Avionics (2 week)	1/20/19	1/27/19	07:00	001D87	DEN	OPC	Musk, Elon	
	A456789	B757 Systems Avionics (2 week)	1/20/19	1/27/19	07:00	001D87	DEN	OPC	Musk, Elon	



Screen Mockup: Exceptions

The screenshot displays the 'Learning Scheduling System' interface. On the left is a dark blue navigation sidebar with the 'UNITED' logo and menu items: HOME, CALENDAR, EXCEPTIONS (highlighted), SCHEDULE, COURSE CATALOG, and WHO CAN TEACH. The main header area includes the system name 'Learning Scheduling System' and the user name 'Kailash Saravanan'. A blue 'Add Exception' button is located in the top right. Below the header is a search and filter section with dropdown menus for 'From Date', 'To Date', 'Instructor', 'Code', 'Point', 'Cancelled', and 'Status'. The central part of the screen features a table titled 'Exceptions' with the following columns: Action, Supervisor, Instructor, Exceptions, Start Date, End Date, and Comment. The table contains 9 rows of data, all representing 'VAC: Vacation' exceptions requested on 1/27/2019 at 07:23:36 PM.

Action	Supervisor	Instructor	Exceptions	Start Date	End Date	Comment
	Mr. Supervisor	John, Johnson	VAC: Vacation	1/20/19	1/27/19	Requested: 1/27/2019 07:23:36 PM
	Mr. Supervisor	John, Johnson	VAC: Vacation	1/20/19	1/27/19	Requested: 1/27/2019 07:23:36 PM
	Mr. Supervisor	John, Johnson	VAC: Vacation	1/20/19	1/27/19	Requested: 1/27/2019 07:23:36 PM
	Mr. Supervisor	John, Johnson	VAC: Vacation	1/20/19	1/27/19	Requested: 1/27/2019 07:23:36 PM
	Mr. Supervisor	John, Johnson	VAC: Vacation	1/20/19	1/27/19	Requested: 1/27/2019 07:23:36 PM
	Mr. Supervisor	John, Johnson	VAC: Vacation	1/20/19	1/27/19	Requested: 1/27/2019 07:23:36 PM
	Mr. Supervisor	John, Johnson	VAC: Vacation	1/20/19	1/27/19	Requested: 1/27/2019 07:23:36 PM
	Mr. Supervisor	John, Johnson	VAC: Vacation	1/20/19	1/27/19	Requested: 1/27/2019 07:23:36 PM
	Mr. Supervisor	John, Johnson	VAC: Vacation	1/20/19	1/27/19	Requested: 1/27/2019 07:23:36 PM



Screen Mockup: Calendar

The screenshot shows a web browser window displaying the 'Learning Scheduling System' interface. The browser's address bar is empty. The page header includes the 'UNITED' logo on the left, the title 'Learning Scheduling System' in the center, and the user name 'Kailash Saravanan' on the right. A dark blue sidebar on the left contains a menu with the following items: HOME, CALENDAR (highlighted), EXCEPTIONS, SCHEDULE, COURSE CATALOG, and WHO CAN TEACH. The main content area features a row of seven dropdown menus for filtering: Month, Course, Catalog, Point, Location, Supervisor, and Instructor. Below these is a calendar for the month of January, displayed as a grid with dates from 30 to 2. At the bottom of the calendar area is a 'Calendar Type' dropdown menu.

UNITED

Learning Scheduling System

Kailash Saravanan

HOME

CALENDAR

EXCEPTIONS

SCHEDULE

COURSE CATALOG

WHO CAN TEACH

Month Course Catalog Point Location Supervisor Instructor

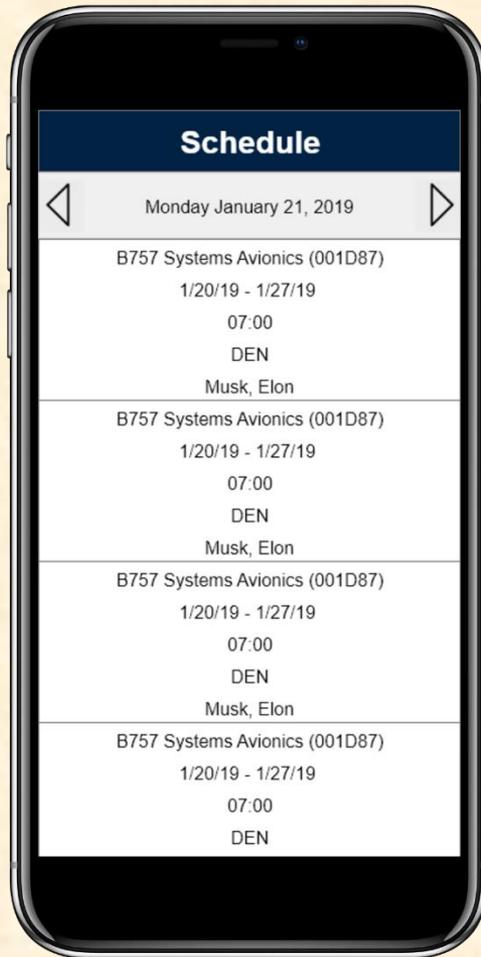
January

30	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

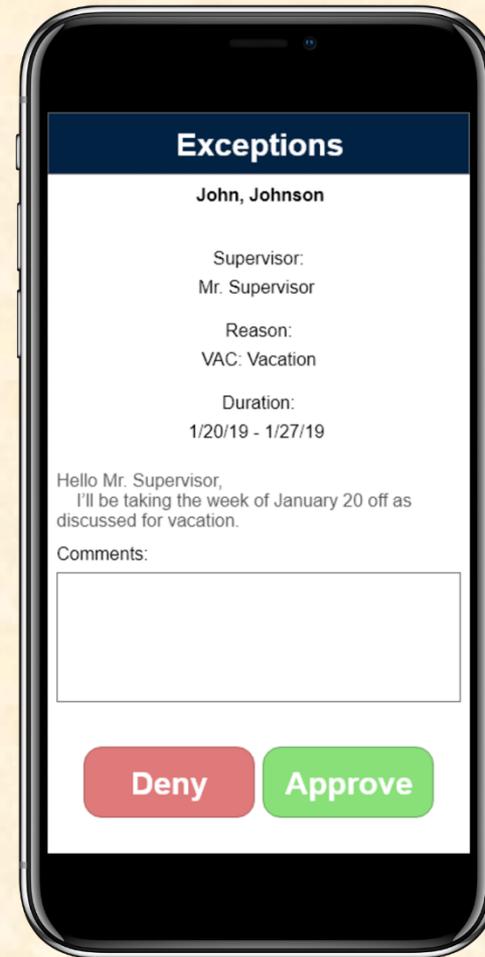
Calendar Type



Screen Mockup: iOS Schedule



Screen Mockup: iOS Exceptions

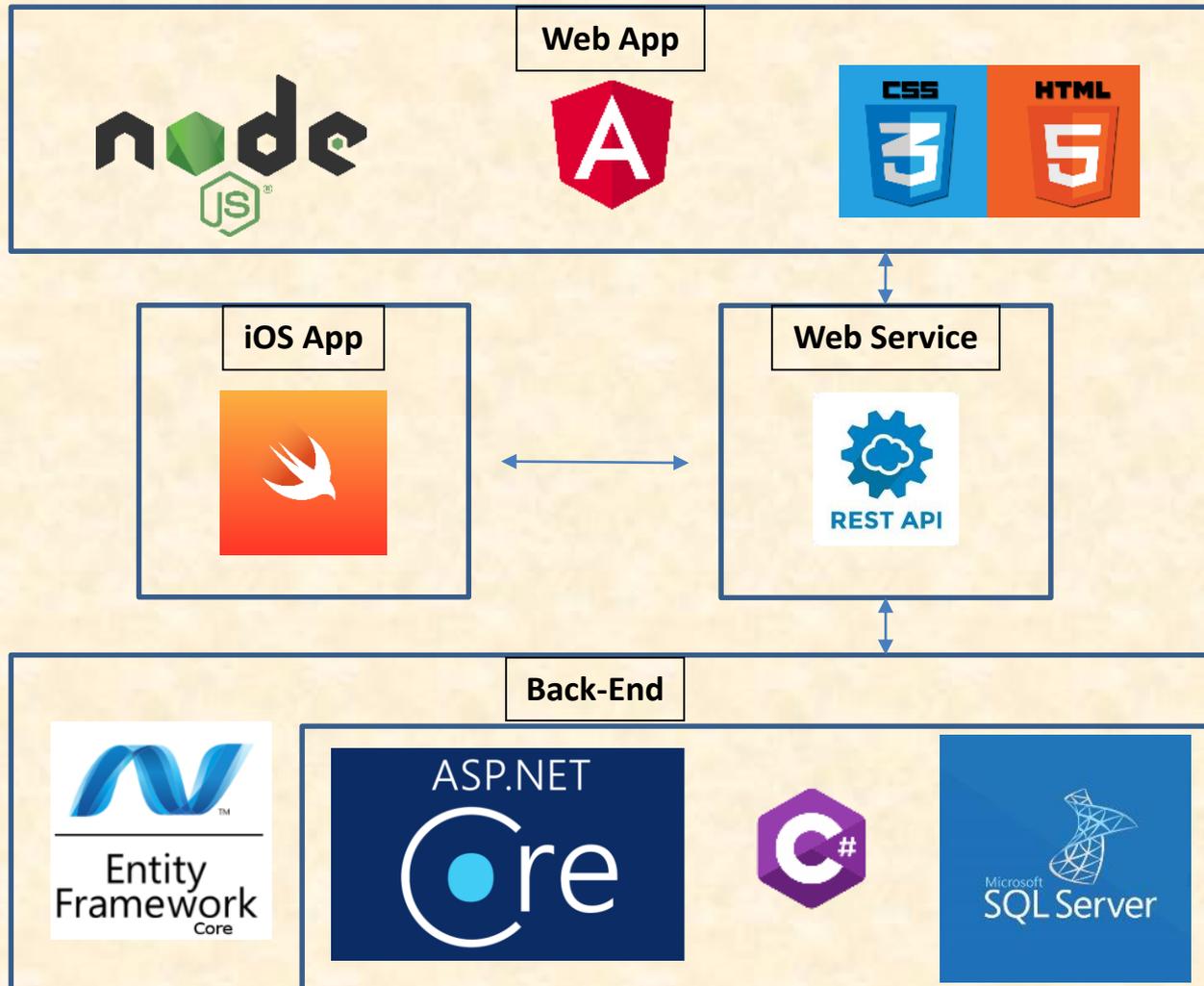


Technical Specifications

- Web App:
 - An ASP.NET Core back-end using Entity Framework will retrieve and process data from the database
 - Present the data with Angular 7 and Bootstrap 4.2 on front-end
- iOS App:
 - Using XCode storyboards to create UI
 - Utilize Swift in combination with MVC
- REST API:
 - Connection from both iOS and web app to the back-end
- Microsoft SQL Server:
 - Allows for storage and retrieval of all data



System Architecture



System Components

- Hardware Platforms
 - PC / Mac
 - iOS devices (iPhone and iPad)
 - Rack server for database

Software Platforms / Technologies

- Visual Studio 2017 Community Edition
- XCode 10
- DayPilot Calendar Plugin
- Angular 7, Node.js, Entity Framework, REST API, ASP.NET Core, C#, Swift, Microsoft SQL Server 2016



Risks

- Consistent user experience
 - It is important to minimize the effort of the users to become familiar with the new system
 - Frequent demonstrations of the prototypes to the client to ensure a smooth transition from the old platform to the new
- Excel spreadsheet conversion
 - United Airlines currently uses a complex excel spreadsheet system to determine the locations of the courses. It is unknown if it is possible to decompose the excel sheets in an automatic process
 - Develop prototype of the automated process and work with the current United Airlines scheduler to ensure desired functionality
- Web service connection
 - iOS prevents connections to sites that are not secure (https) without an SSL certificate
 - Contact university's website services department to determine whether we can get approved in the MSU SSL certification program. If not, reach out to professors with subject matter knowledge for assistance
- Accuracy of the mock data
 - The mock data we generated may result in improper feature developments because of possible inaccuracies in the data itself
 - Confirm with the client that our mock data is a good representation of the current data in the existing system to guarantee features are developed as intended



Questions?

?

?

?

?

?

?

?

?

?

