

Companion File Browser

Team Name: SDDEC19-11

Client: Kirit Chandran

Adviser: Dr. Zheng

Team Members:

Christopher Bui, Brian Chodur, Luke Stoll, Bei Zhao, Zhenyu Zhao

<https://sddec19-11.sd.ece.iastate.edu>

Introduction

About our client

- ▶ Buildertrend provides a cloud-based construction management software
- ▶ Service includes web and mobile applications for both Android and iOS
- ▶ #1 Software for home builders, contractors and remodelers

Problem Statement

Buildertrend wants a new sub-system extension for their current software product which is a web application

- ▶ Cloud file sharing system that allows users to sync and download files on the cloud
- ▶ Provide users easy collaborations, error recovery, and high accessibility

Project Benefits

- ▶ More Security
- ▶ Custom Maintenance
- ▶ Lower Cost

Market Survey

Similar Products:

- ▶ Dropbox
- ▶ Google Drive
- ▶ OneDrive
- ▶ Box

Requirements

Functional Requirements

- ▶ A web application that has following:
 - ▶ Create users and assign users' permission levels
 - ▶ Upload, download files and browse files' information
 - ▶ View files' history and download different version files
- ▶ Expand the capabilities of the btconnect application:
 - ▶ Real-time files and folders sharing with web application
 - ▶ Real-time notifications
- ▶ Web API Interface that btconnect app could connect with to send and receive data

Non-functional Requirements

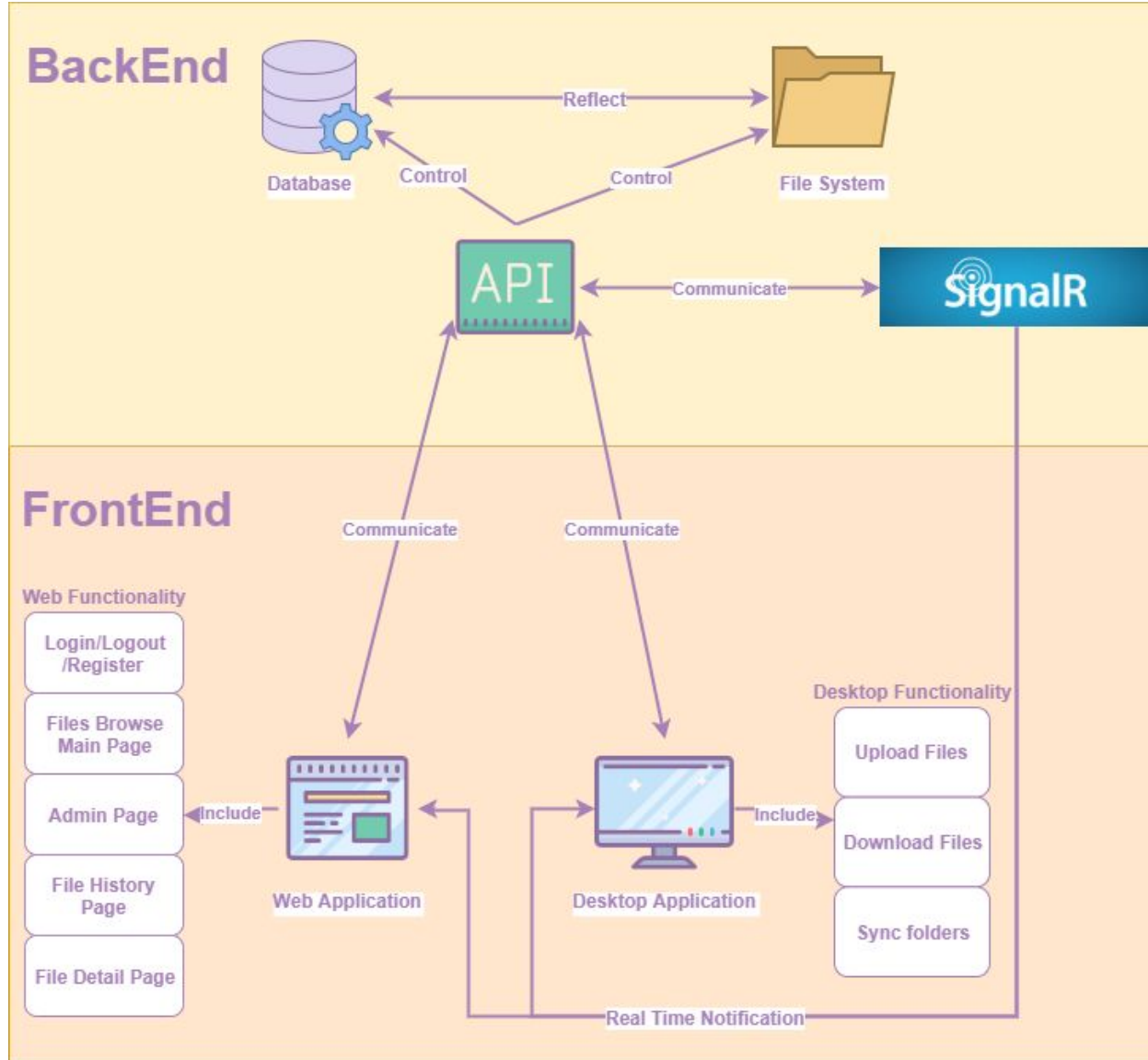
- ▶ **Maintainability** - Code and documents should be easy for future developers to understand.
- ▶ **Usability** - The application should be user-friendly and easy to use without a detailed explanation about how the application works.
- ▶ **Security** - The documents and user information uploaded into the system should only be accessed by people who have permission.

System Design

System Constraints

- ▶ Labor Resource and Time
- ▶ Required use of technology:
 - ▶ React.js frontend library
 - ▶ SignalR to communicate between Restful API and applications
 - ▶ C# & .Net Framework for API
- ▶ Integration with Pre-existing Application

Overview of Design



System Analysis

- ▶ Backend
 - ▶ Database - Microsoft SQL Server
 - ▶ Security- Json Web Tokens
 - ▶ Real Time Notifications - SignalR
 - ▶ Web API - .Net Framework
- ▶ Frontend
 - ▶ Library- React.js
 - ▶ Rendering Technique - Client side rendering

Functional Decomposition

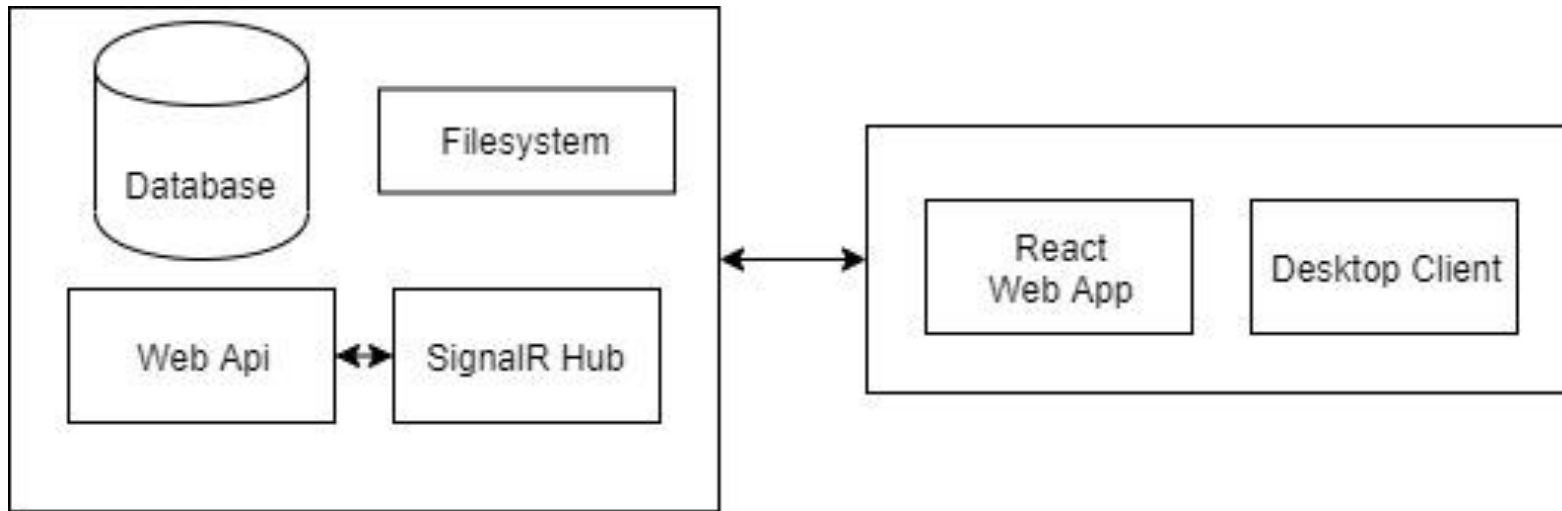
- ▶ User signs into web application
- ▶ User can view/download/upload files
- ▶ User can access the same information on the desktop application

Detailed Design Implementation

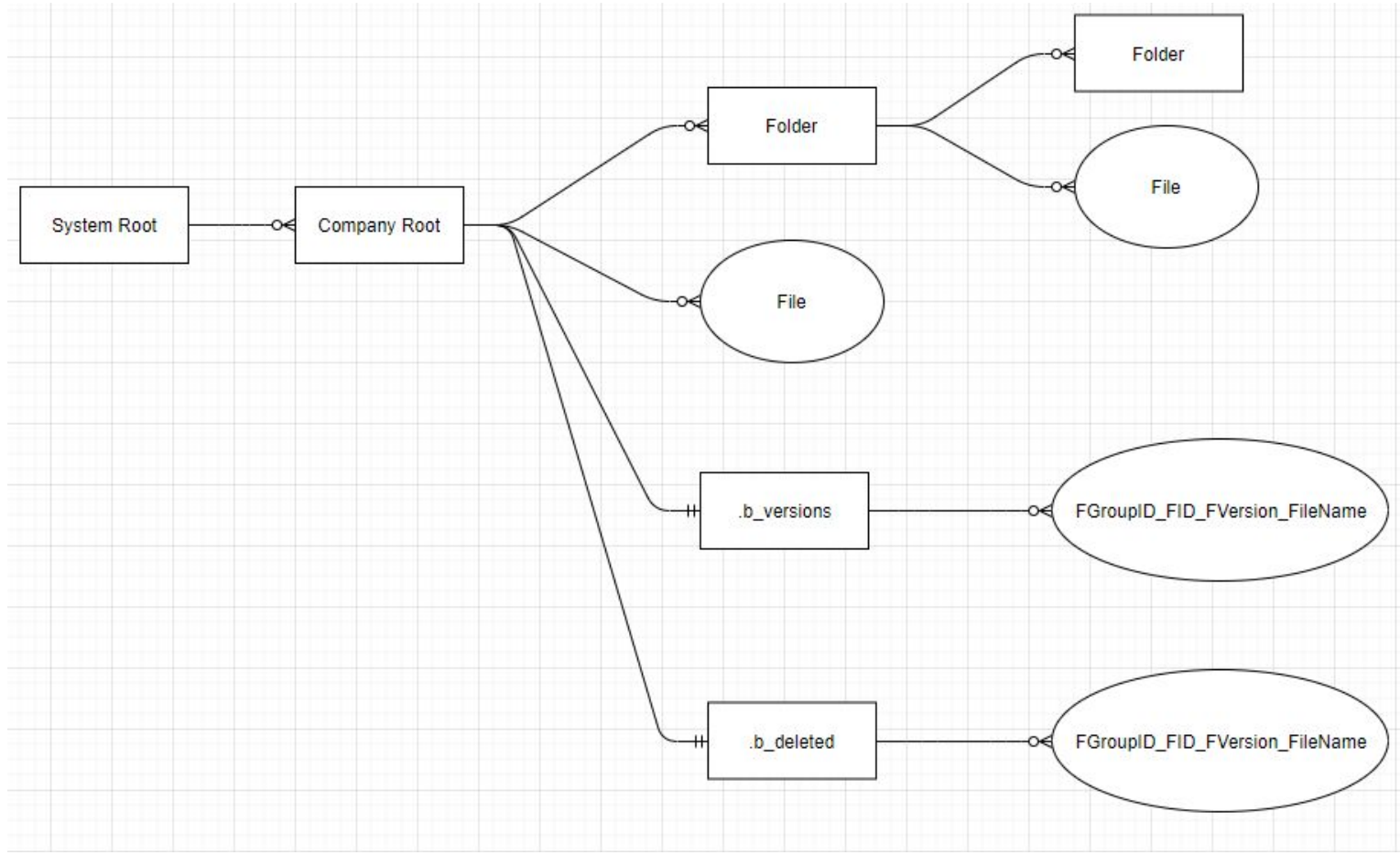
Software Platforms

- ▶ React for web UI
- ▶ C# for backend services and web API
- ▶ Microsoft SQL Server as database.
- ▶ No design tradeoffs as we were limited to what we could use

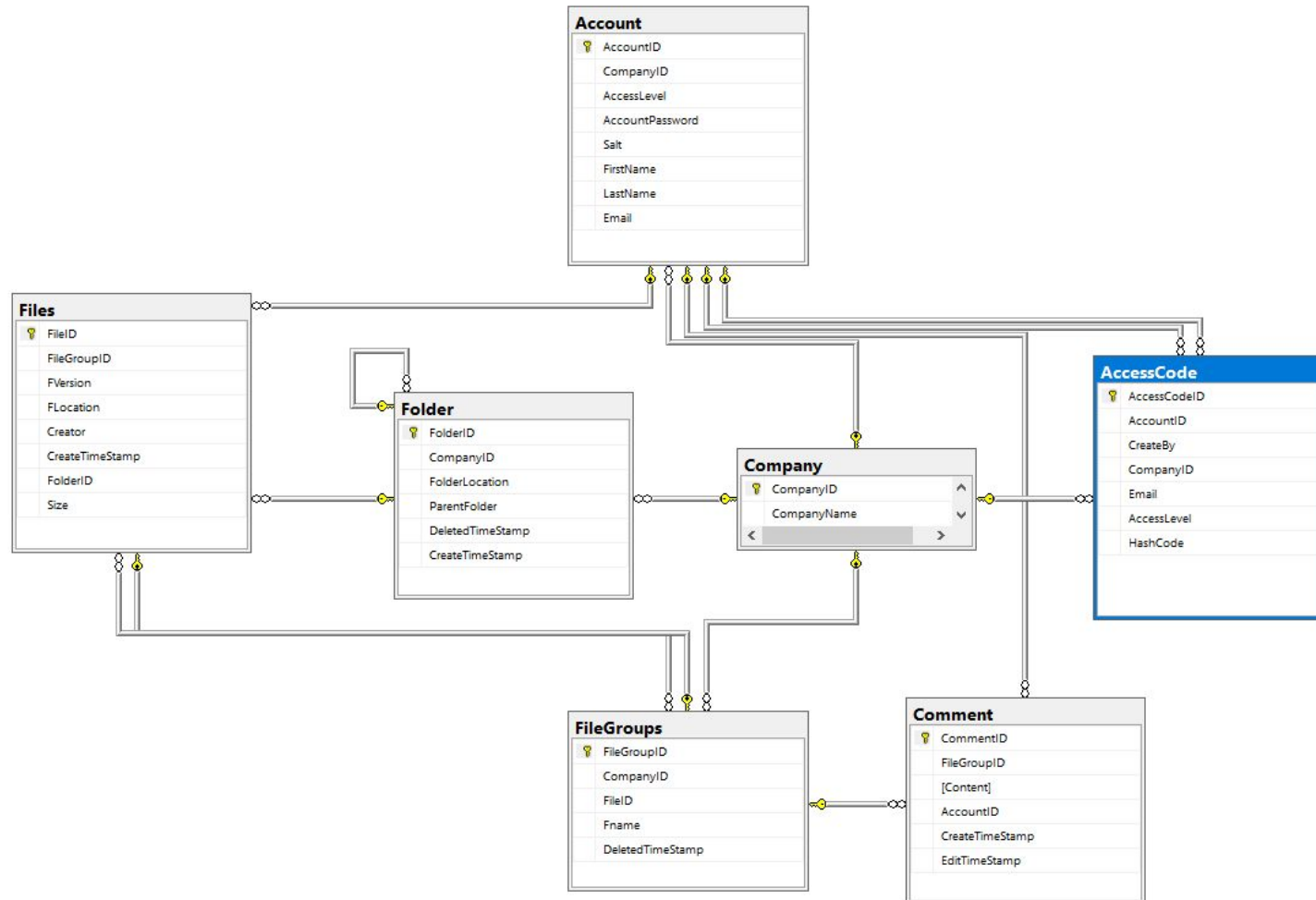
Detailed High Level Design - Functional Modules



File System Design - Final Iteration



Database Design - Final Iteration



Interface Design

- ▶ User Interface:
 - ▶ Web page that is built on React.js
- ▶ Communication Interface:
 - ▶ Http Post/Get Request through rest api
 - ▶ Sockets for real time notifications

User Interface Design

- ▶ Home Page
 - ▶ Main component to list folders, files, and respective actions
- ▶ Admin Panel
 - ▶ Able to create new code for new user
 - ▶ Disable user account
 - ▶ Set/Change permission

Desktop application

- ▶ Items/requirements completed
 - ▶ Login
 - ▶ Set up a sync folder
 - ▶ Sync down folders and empty placeholder files
 - ▶ Download/Upload files within Windows File Explorer
- ▶ Items/requirements unfinished
 - ▶ Delete/Recover/Revert files
 - ▶ Create/Delete/Recover/Revert folders
 - ▶ File edit warning notification

Testing

Testing Plan

- ▶ Integration Testing
 - ▶ Postman - API endpoint, database, filesystem
 - ▶ Mock Data - React component rendering
- ▶ System Testing
 - ▶ Manual test - end to end testing web app, api, desktop client

Project & Risk Management

Project Milestones & Schedule

TASK NAME	ASSIGNED TO	START DATE	DUE DATE
Gather Requirement		2/12/2019	2/13/2019
System/Architecture Design		2/17/2019	2/24/2019
Define Scope		2/25/2019	3/4/2019
Define Edge Case		3/5/2019	3/18/2019
Define Test Case		3/5/2019	3/18/2019
Web Frontend UI Design		3/25/2019	5/5/2019
Backend API Design		3/25/2019	5/5/2019
Desktop Application UI Design		3/25/2019	5/5/2019
Database/File System Design		3/25/2019	5/5/2019
Web Interface Setup		8/20/2019	9/9/2019
Database Development		8/20/2019	9/9/2019
Server Backend Development		8/20/2019	9/9/2019
Front and Backend Connectivity		9/10/2019	9/26/2019
Desktop Application Extension		9/27/2019	10/11/2019
Desktop Application Connectivity		10/13/2019	10/27/2019
Documentation		10/28/2019	11/11/2019
Functional Testing		11/12/2019	11/26/2019
Non-Functional Testing		11/26/2019	12/10/2019

TASK NAME	ASSIGNED TO	START DATE	DUE DATE	DURATION
Gather Requirement		2/12/2019	2/13/2019	1
System/Architecture Design		2/17/2019	2/24/2019	7
Define Scope		2/25/2019	3/4/2019	7
Define Edge Case		3/5/2019	3/18/2019	13
Define Test Case		3/5/2019	3/18/2019	13
Web Frontend UI Design		3/25/2019	5/5/2019	41
Backend API Design		3/25/2019	5/5/2019	41
Desktop Application UI Design		3/25/2019	5/5/2019	41
Database/File System Design		3/25/2019	5/5/2019	41
Web Interface Setup		8/20/2019	9/9/2019	20
Database Development		8/20/2019	9/9/2019	20
Server Backend Development		8/20/2019	9/9/2019	20
Front and Backend Connectivity		9/10/2019	12/2/2019	83
Desktop Application Extension		11/25/2019	12/11/2019	16
Desktop Application Connectivity		11/27/2019	11/28/2019	1
Documentation		12/2/2019	12/9/2019	7
Functional Testing		11/12/2019	11/26/2019	14
Non-Functional Testing		11/26/2019	12/10/2019	14

File Companion Browser - SDDEC19-11

Risk Identification

- ▶ Compatibility
 - ▶ Server easily interface w/ existing desktop application
- ▶ Delivery Time
 - ▶ Features/Parts might not get delivered due to time constraint/technical issue/software bugs.
- ▶ Lack of knowledge on React, Redux, .Net
 - ▶ Both semesters spent learning and reading documentation

Risk Mitigation

- ▶ Both semesters spent learning and reading documentation
- ▶ Trello board for task management
- ▶ Continuous testing and communication
- ▶ Changing system design during development
 - ▶ Constant communication to discuss the changes, effects, and outcomes

Demo

File Companion Browser - SDDEC19-11

Lessons Learned

- ▶ No matter how much planning is done, issues will still occur
 - ▶ Edge cases while developing we didn't account for
 - ▶ Bugs taking longer than expected
- ▶ Communication is key
 - ▶ Who is working on what task
 - ▶ Who needs help to break roadblocks

Conclusion

Project Status

- ▶ Web Application
 - ▶ Full functionality
- ▶ Desktop Application
 - ▶ Sync Folder
 - ▶ Upload/Download files
 - ▶ Notifications

Future work

- ▶ Final code and documentation hand-off to BuilderTrend
- ▶ BuilderTrend will need to implement file editing and call our API endpoints as documented

Questions?

Thank you