Image Management Service

Bohorquez, Juan Bohorquez, Tomas Buch, Camila Chen, Davis
Garcia, Kevin Howie, Roy Luu, Steven Smelser, William Vonk, Kelsey

Contents

1 System Analysis **4**

1.1 System Overview 4

1.2 System Diagram 4

1.3 Actor Identification 4

1.4 Design Rationale 4

1.4.1 Architectural Style 4

1.4.2 Design Patterns 4

1.4.3 Framework 4

2 Functional Design **5**

3 Structural Design **8**

List of Figures

1 Search and Download Use Case 5
2 Sequence Diagram 6
3 Admin Upload Diagram 6
4 Admin Track Diagram 7
5 Admin Remove Diagram 7

List of Tables

1 System Analysis

1.1 System Overview

The System will be made up of four main parts: the Database, the user interface, the controller, and a file watcher. The Database will consist of multiple projects (these projects contain pictures, text files and/or videos) that span across various dates and locations. The user interface gives the user the ability to query trough the database by date, project name, and various other paramiters. The controller will take the parameters the user gives and convert them into somthing the database can use to find the requested files, after the database finds the files the controller then tells the UI what the files are so that it can display them. The file watcher will update the database whenever a file is added or removed.

The type of architecture used for the application is a mix between the MVC (model view controller) and MVVM (model view view-model) styles. The models sit in the MongoDB database; Meteor will serve as the controller, and React will be used for the view.

1.2 System Diagram

1.3 Actor Identification

1.4 Design Rationale

1.4.1 Architectural Style

1.4.2 Design Patterns

1.4.3 Framework

2 Functional Design

Figure 1: Search and Download Use Case

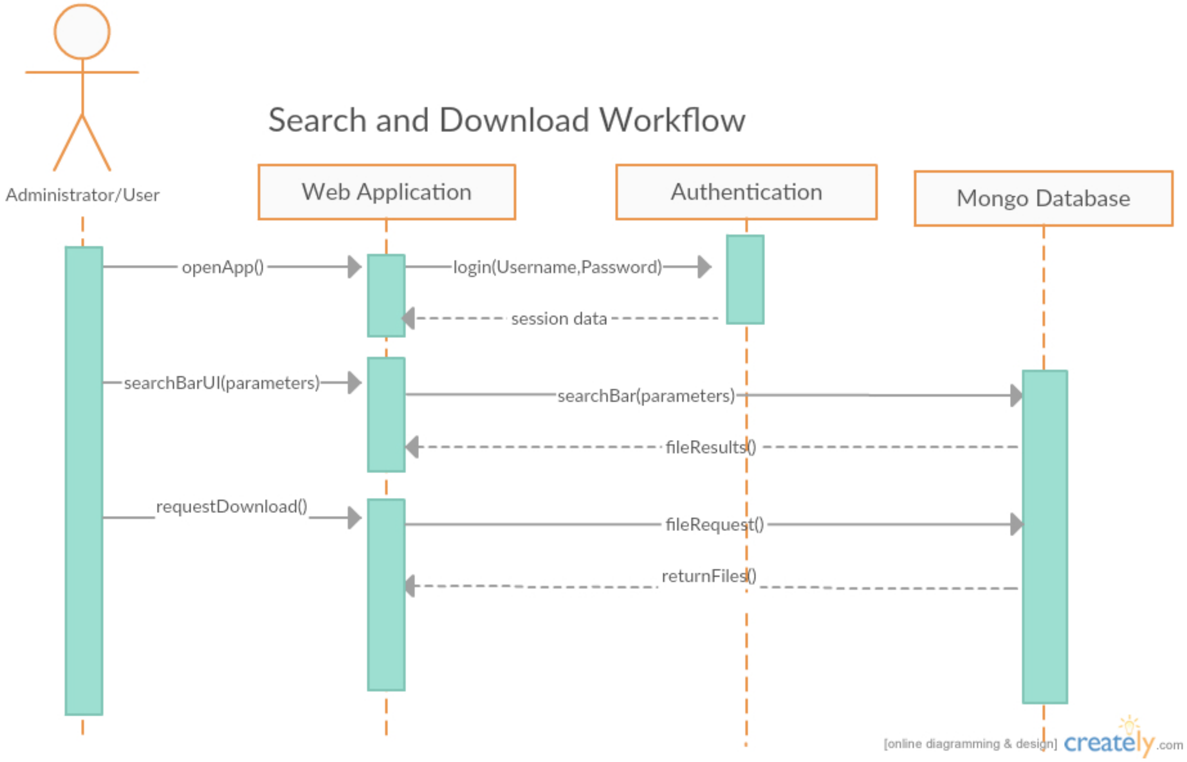


Figure 2: Sequence Diagram

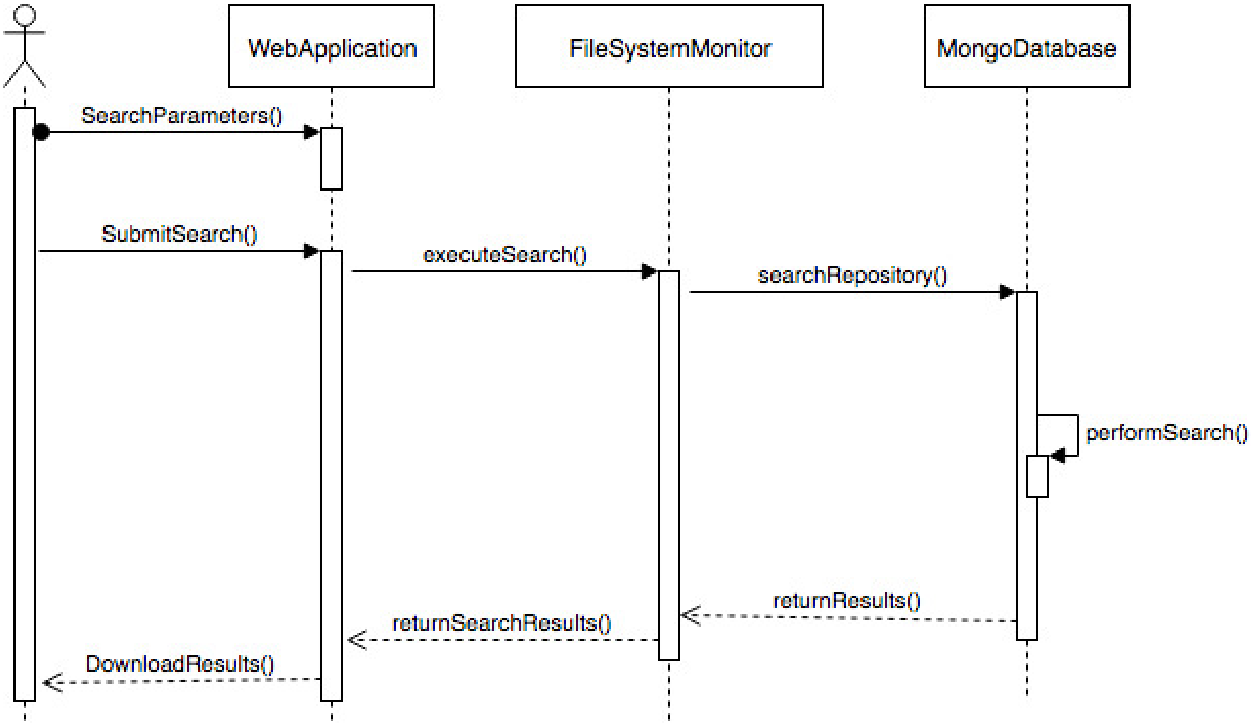


Figure 3: Admin Upload Diagram

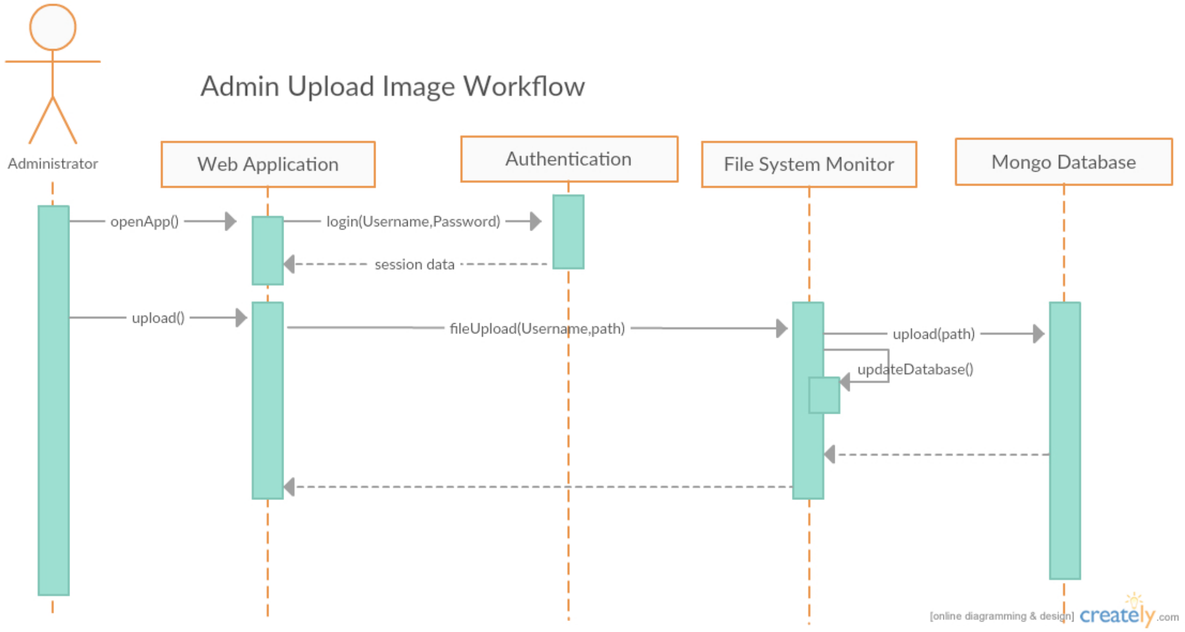


Figure 4: Admin Track Diagram

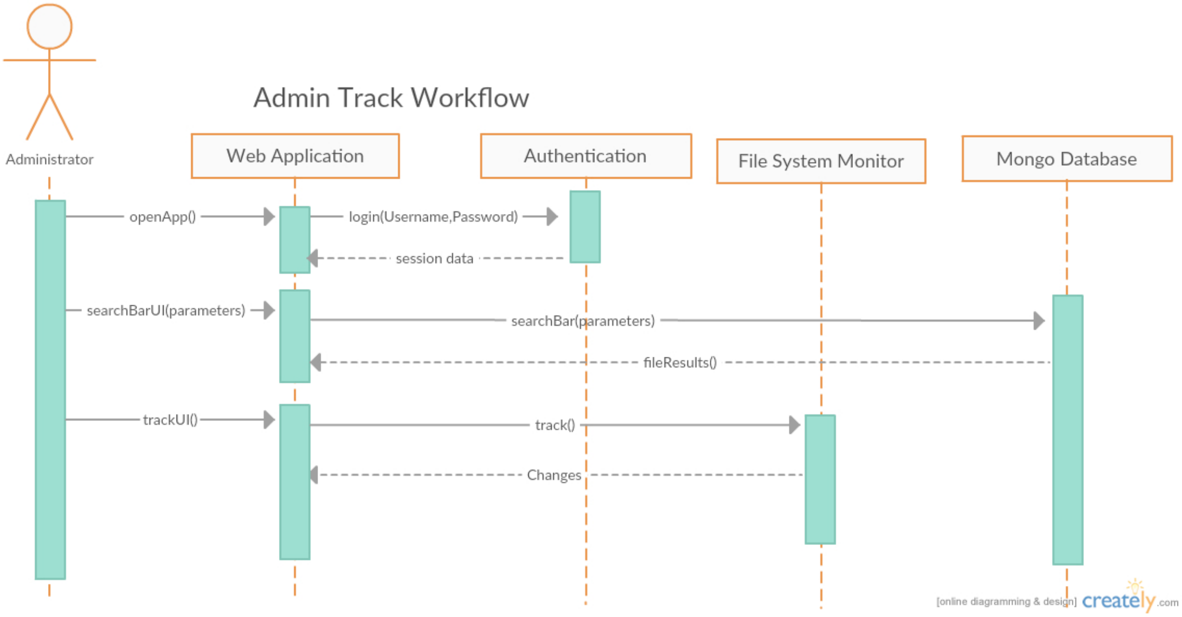
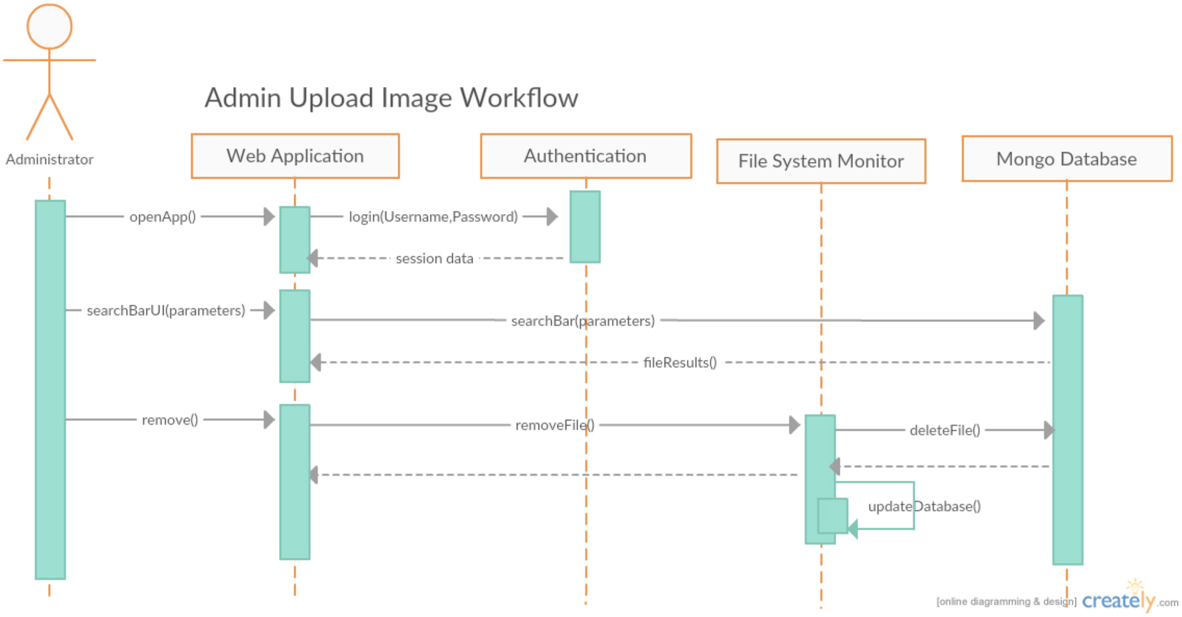


Figure 5: Admin Remove Diagram



3 Structural Design