[redacted]

Software Requirement specification
(Kneat Coding Challange Assignment)

Author - Makrand Vichare

Dated: 23-Sept-2017

TABLE OF CONTENTS

0 PREFACE

0.1 PURPOSE OF THIS DOCUMENT

This document is a ***Software Requirement Specification*** *for understanding the software requirement and application architecture of the* ***coding challenge assignment project for Kneat****. It provides guidance which is intended to assist the relevant management or technical staff, whether client or supplier, in project development and deployment. It is also useful background reading for anyone involved in developing or monitoring the assignment.*

0.2 SPECIFIC DESIGN CONSIDERATIONS

1. *The specification is developed for the dotnet technologies like* ***Asp.net MVC****,* ***Asp.net web api2*** *and client side technologies likes* ***angularjs*** *and* ***typescript****.*
2. *The Unified Modelling Language (****UML****) paradigm is used to develop design diagrams, which is rapidly becoming recognised as an industry standard in software development.*

1 INTRODUCTION

This document provides a complete software requirement and application architecture.. The application is a web based and can be accessed with proper access rights. The application will be used for calculating starships travel and resupply data..

1.1 PURPOSE

a. *The purpose of the document is to provide a design guidelines while developing the application. The document will give complete details of the application design, assumptions and will assist client and relevant teams.*

1.2 SCOPE

a. *Deliver a web application which will be used by starship pilots and the starships' captains.*

b. *Deliver a authentication service as a web Api which can be consumed and accessed by any device for user authentication.*

c. *StarShips and planets data will be retrieved from third party web api* ***https://swapi.co/*** *which is public facing and free to use.*

1.3 DEFINITIONS, ACRONYMS AND ABBREVIATIONS

a. *The following is a list of definitions for this template based on the UML approach to system design:*

b.

Class Diagram	*Describes the structure of a system*
Activity Diagram	*Describes activities and actions taking place in a system*
Sequence Diagram	*Shows one or several sequences of messages sent among set of objects*

2 SOFTWARE REQUIREMENTS

2.1 Background*:*

As part of this code challenge you will be using an API available here: https://swapi.co/

We want to know for all SW star ships, to cover a given distance, how many stops for resupply are required.
The application will take as input a distance in mega lights (MGLT).
The output should be a collection of all the star ships and the total amount of stops required to make the distance between the planets.

All other application details are at your own discretion.

Sample output for 1000000 input:
Y-wing: 74
Millennium Falcon: 9
Rebel Transport: 11

NOTE: The console application can be created in any language you wish to use (not limited to .NET languages).

Requirements
1) The completed code should be submitted along with
2) Accompanying documentation
3) Tests and instructions on the usage of the application.

If there are any queries on the challenge I can be contacted for clarification if necessary.
All aspects of the challenge will be considered during the review, coding style, code organization, correct calculations, working application etc.
This is a very important part of the interview process and we would like our candidates to put their best code forward.

3 SYSTEM OVERVIEW

3.1 SYSTEM ARCHITECTURE

1. *The application is N layer application which has the following layers.*

   UI layer *- it facilitates access by both external and internal clients through technologies such as web browsers. This is developed using the technologies like* ***asp.net MVC, angular js and typescript****.*

   Web Api Service layer *- it authenticate user and provide access to the resource server. This layer allows to integrate with various devices e.g. web browser, mobile, desktop app. This is developed using* ***asp.net web api 2*** *which provides* ***token based authentication*** *and can be scaled easily. This layer interacts with underlying domain service layer.*

   Third Party Service layer *- it provides the starships. planets and other required info.*

3.2 INFRASTRUCTURE SERVICES

3. *The Infrastructure Services can be built incrementally, implementing to handle the following concerns -*

   a. ***Security*** *– resource server is secured by implementing authorization server and the gated resource cannot be accessed until users are authenticated. In the future need to implement different roles with various access rights.*

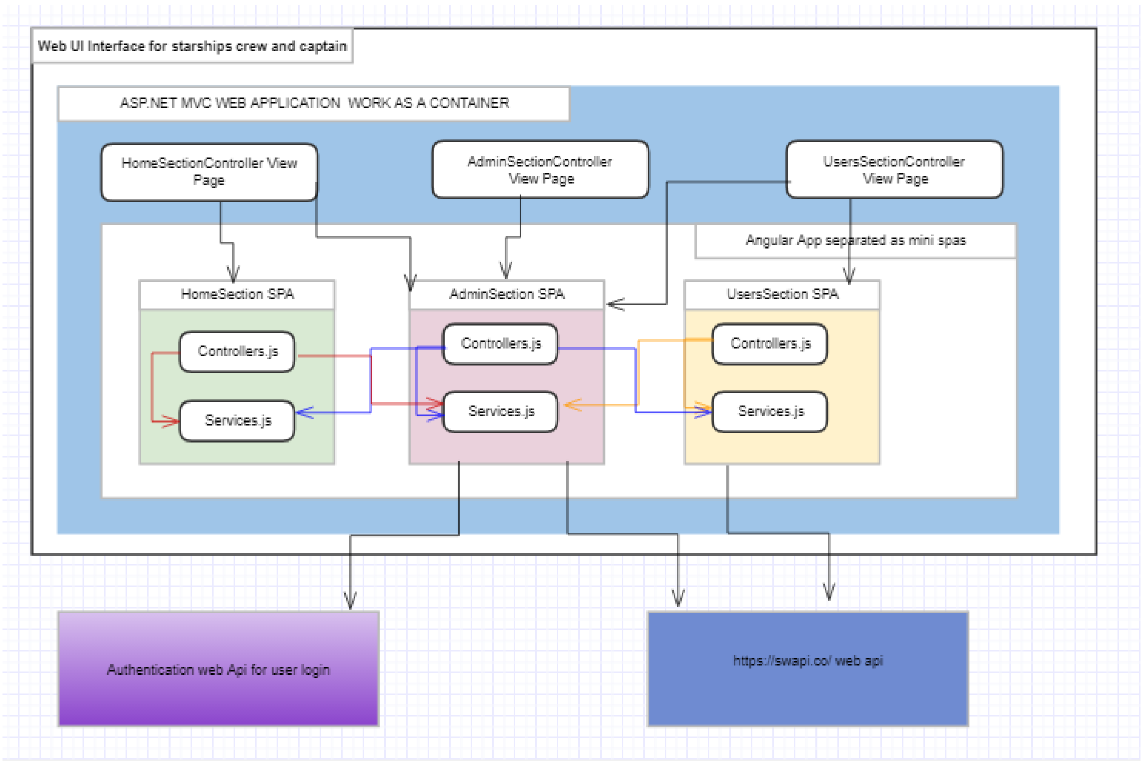
   b. ***Performance monitoring and reporting*** *– not yet implemented*

   c. ***Error Handling*** *– handled limited way.*

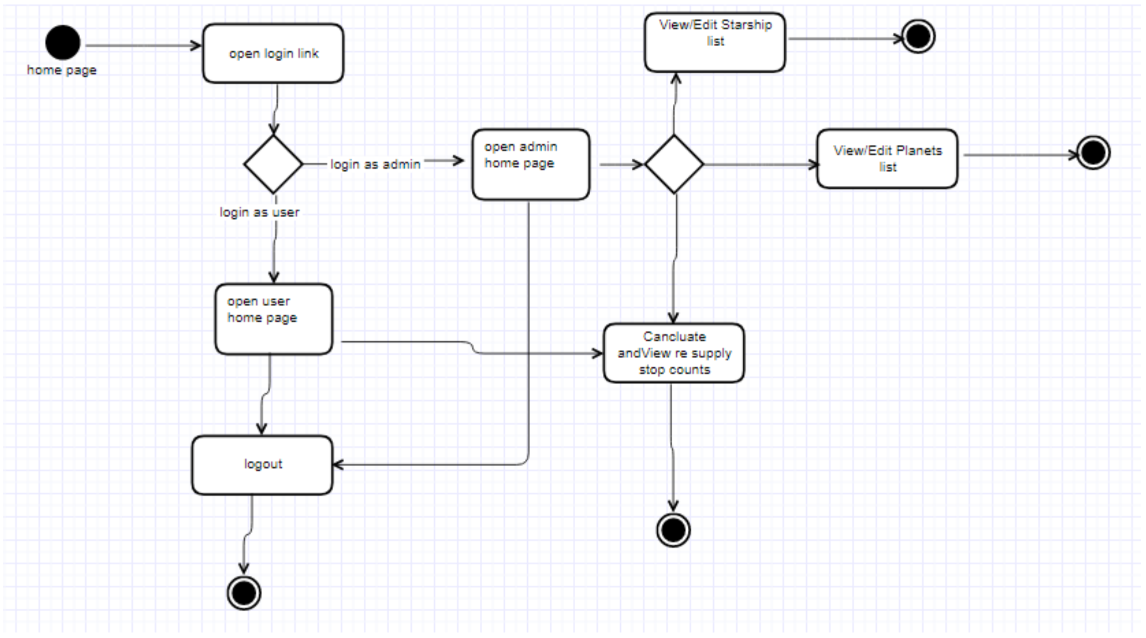
   d. ***Debugging & logging*** *– infrastructure layer is available and can used to store all debug and log info.*

4 ARCHITECTURE VIEWS

4.1 COMPONENT DIAGRAM AND LAYER DIAGRAM (merged to save time)

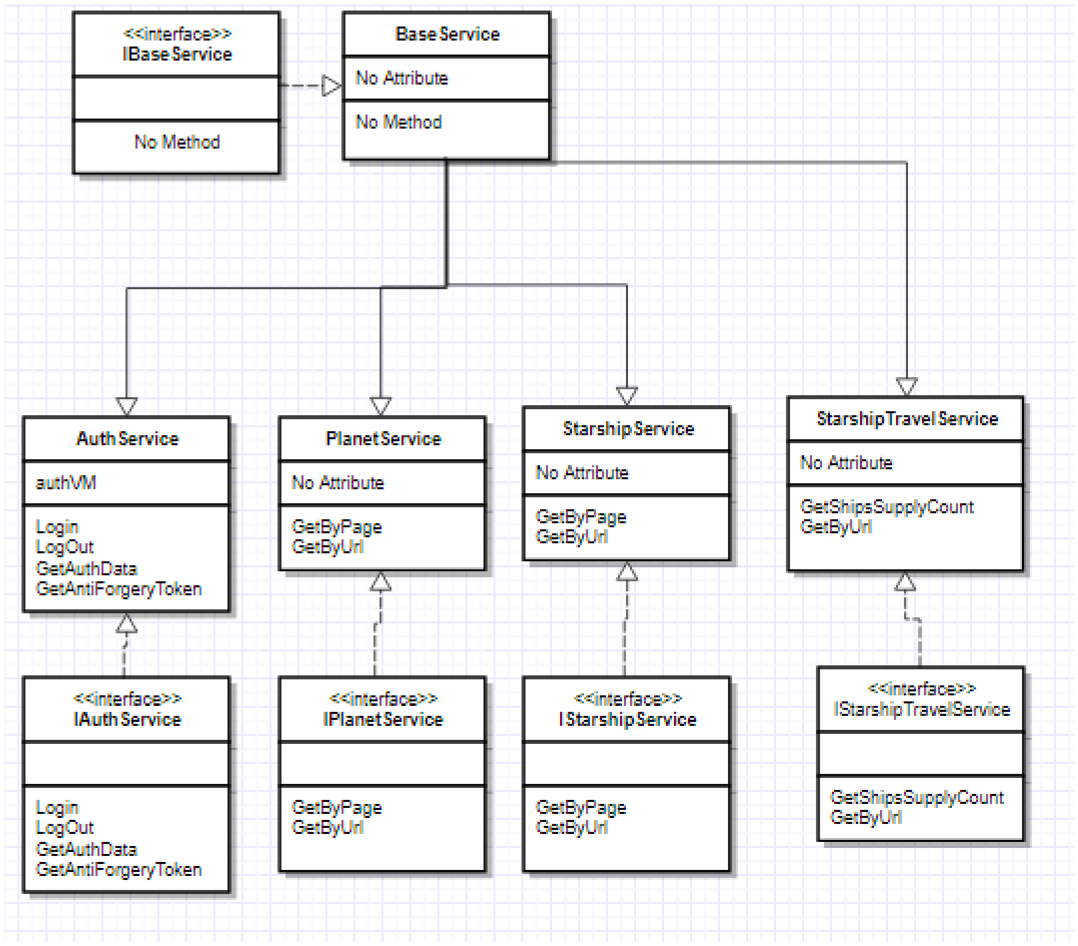

Web UI Interface for starships crew and captain
ASP.NET MVC WEB APPLICATION WORK AS A CONTAINER
HomeSectionController View Page
AdminSectionController View Page
UsersSectionController View Page
Angular App separated as mini spas
HomeSection SPA
Controllers.js
Services.js
AdminSection SPA
Controllers.js
Services.js
UsersSection SPA
Controllers.js
Services.js
Authentication web Api for user login
https://swapi.co/ web api

4.2 ACTIVITY DIAGRAM

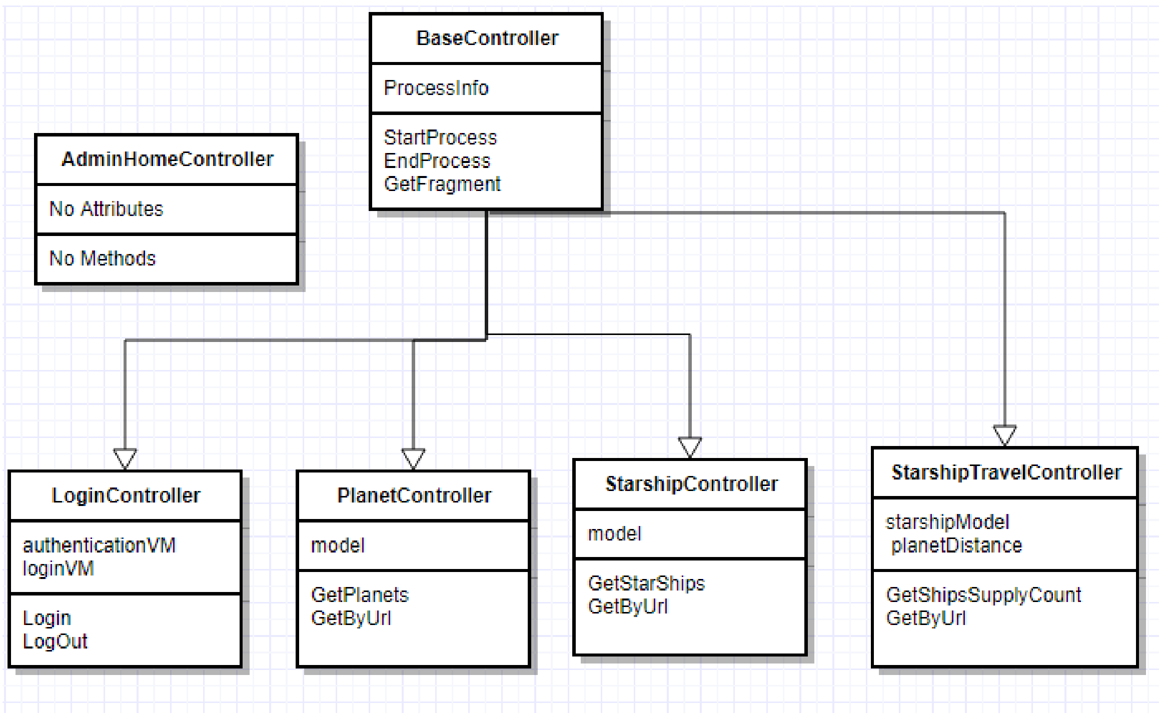


4.3 CLASS DIAGRAM

4.3.1 SERVICE CLASSES :

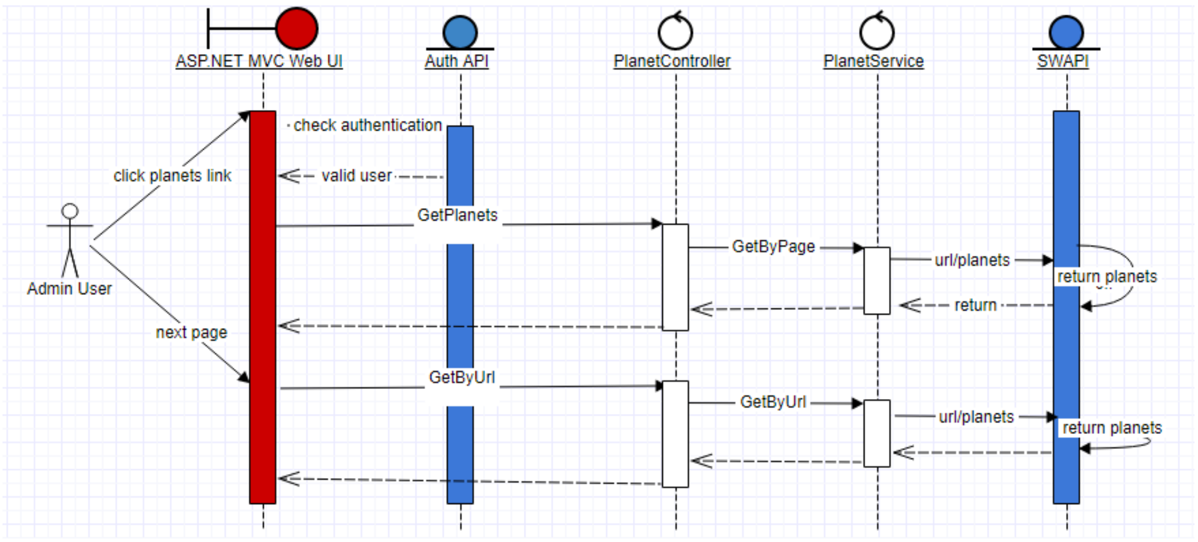


4.3.2 CONTROLLER CLASSES

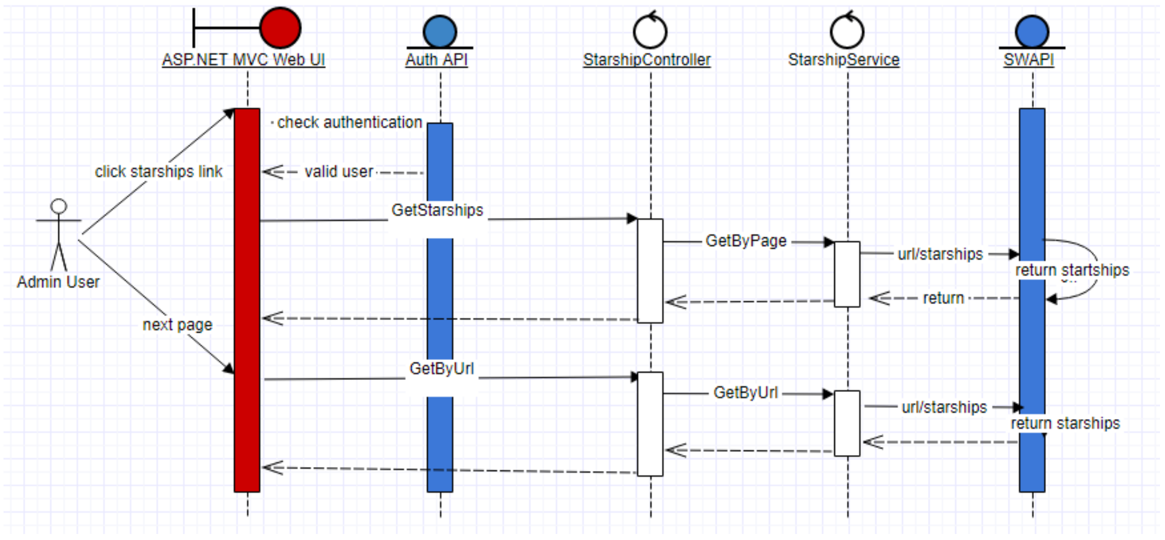


4.4 SEQUENCE DIAGRAM

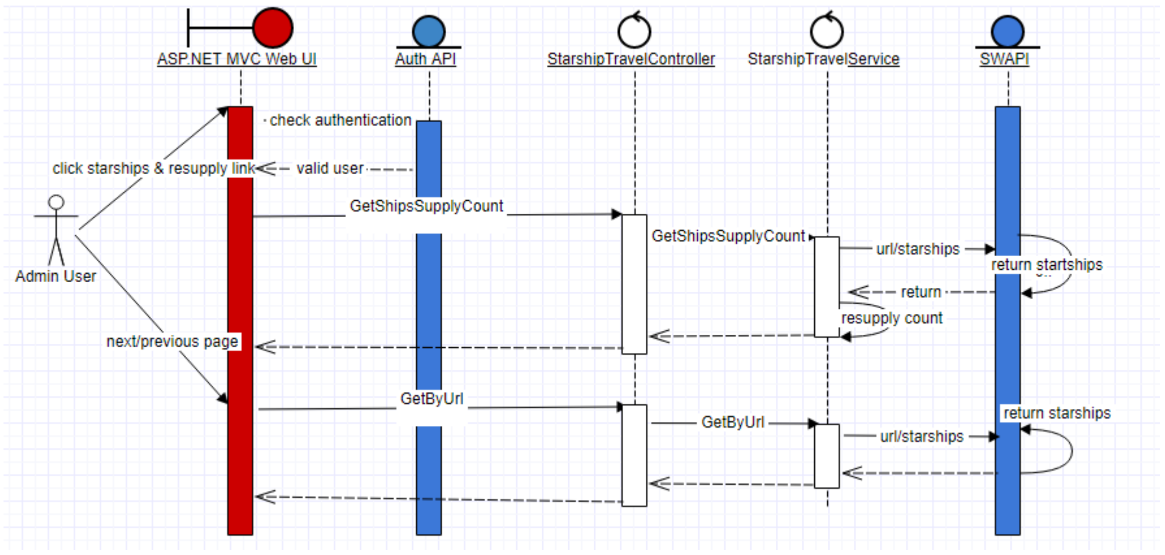
4.4.1 Admin User (Get Planets)



4.4.2 Admin User (Get Starships)



4.4.3 Admin User/Normal User (Calculate Resupply Stops for Starships)



5 BUSINESS RULES & CONSTRAINTS

5.1.1 BUSINESS RULES

a. Only admin user can add or update or delete planets list

b. Only admin user can add or update or delete starships list

c. All users can have access for calculating resupply stops for starships

d. Only authenticated users must have access to this application.

e. Estimated distance can not be zero.

5.1.2 CONSTRAINTS

a. The application depends on starship war api which is third party interface and the application will not work if it is down or not available.

6 DESIGN METHOD AND STANDARDS

The following design patterns are used:

- N- layer pattern, MVC pattern (asp.net MVC)
- MVVM pattern using angularjs with typescript
- Mini Spa Approach for better scalability and maintenance

6.1 SOFTWARE DEVELOPMENT TOOL AND TECHNOLOGIES

Development Environment –

Visual Studio 2017 Community Edition

Asp.net MVC

AngularJs 1.5+

TypeScript 2+

Asp.net Web Api 2

C#

6.2 SOFTWARE DESIGN TOOLS

https://go.gliffy.com