

1 Augmented Chess: Assignment 8

Group Members

Jacob Holm Mortensen	jmorte14@student.aau.dk
Martin Raunkjær Andersen	marand13@student.aau.dk
Thomas Gwynfryn McCollin	tmccol14@student.aau.dk

1.1 Question 1

From the Interface Design Patterns presented in "User interface design practices in simple single page web applications", which ones are used in your application? For those that are used, indicate the concrete implementation used. For those that are not used but that are possible to use, how could they be included to enhance the user interface of your Web application?

Bear in mind that some of the functionality described in this section is not implemented yet, the section describes the finished product or the design.

The application makes additional information available to the user when a piece is selected. This information consists of the pieces attributes as well as the possible valid moves. This is consistent with the details on demand interface design pattern. The pattern has been implemented as a attribute box which shows the currently selected pieces attributes, the box is always visible, but the content changes based on the selected unit. Furthermore the legitimate moves that a piece can make are illustrated as highlighted cells on the chess board, these are only visible when a piece is selected.

When the user is managing their armies their attention is focus using a number of input fields and some confirmation pop-ups. The fields are used to modify the attributes of the army pieces and the pop-ups are used when committing changes. A change could changing the name of an army or finalising the modification of an army.

The application presents the user with all possible actions at any given moment. During the game each piece represents a different course of action and when managing armies, the user is faced with a rich user interface. To counteract increased complexity of this, the application functionality is as simplified as possible. The most complex operations of application are present when managing armies, this interface consists of a large array of different buttons and input

fields. This is consistent with multifunction control.

Finally actions on demand and user driven page component visibility interface patterns could be included in the application. The user driven page component visibility pattern is circumvented by the different dynamic pages generated on the client side. This effectively only shows the relevant information, but this is not directly controlled by the user. The actions on demand pattern could be used instead of the multifunction control, however hiding functionality from the users would not benefit the application in this case.

1.2 Question 2

Consider the advantages and disadvantages of implementing your Web application as a single page application. In which context, would it be better to implement it as a single page application?

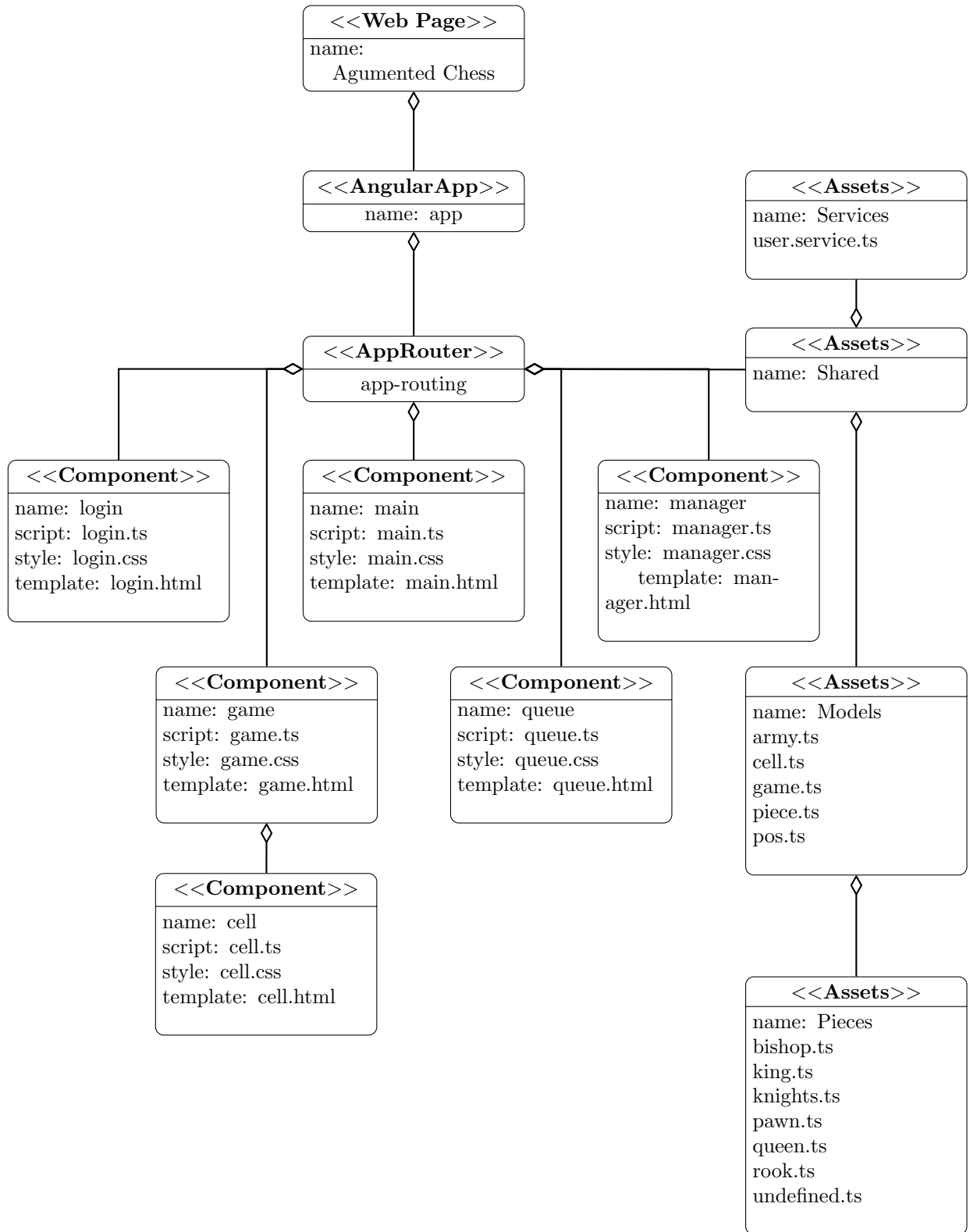
The Augment Chess application is a single page application. The advantages of single page applications outweighs the disadvantages, in the case of Augment Chess. Single page applications allow for more reuse of code then multi page applications, this reduces development complexity. Furthermore, since computation is performed client side, single page applications perform better then multi page applications. This means that the single page application is more "snappy" or fast then a multi page application, which is important for usability and user satisfaction. Furthermore, single page application can use the the same origin pattern to enforce greater security. The reason that this is easier in multi page applications is that the routing component knows all valid sites beforehand, this is the case because valid sites are local or share origin with the application.

The disadvantages of single page applications consist of a large initial download and a much greater memory consumption. The full application must be downloaded initially and the application must be kept in memory throughout use. Furthermore, the single page application only has one page and therefore a single point of failure. Should something go wrong with the page, the entire application is dysfunctional whereas a multi page application could still serve the remaining pages properly regardless of a single broken page.

1.3 Question 3

Use the UML model presented in "A Model-Driven Development of Web Applications Using AngularJS Framework" to model your Web application (or a single page application version of your Web application if you are not implementing a single page application)

The following UML diagram describes the Augment Chess application. The UML is very cluttered, this is due to the size of the diagram and that it has been confined to a single page.



1.4 Question 4

Continue working on the development of your Web application. Report the status of its development.

The application is nearing the end of the first release. The application now supports proper win conditions and most valid moves in chess. All pieces have attributes, which are modifiable, however some edge case attributes are not implemented yet. An example of this is the knight, which as a special movement pattern and therefore has higher development complexity than the other pieces.