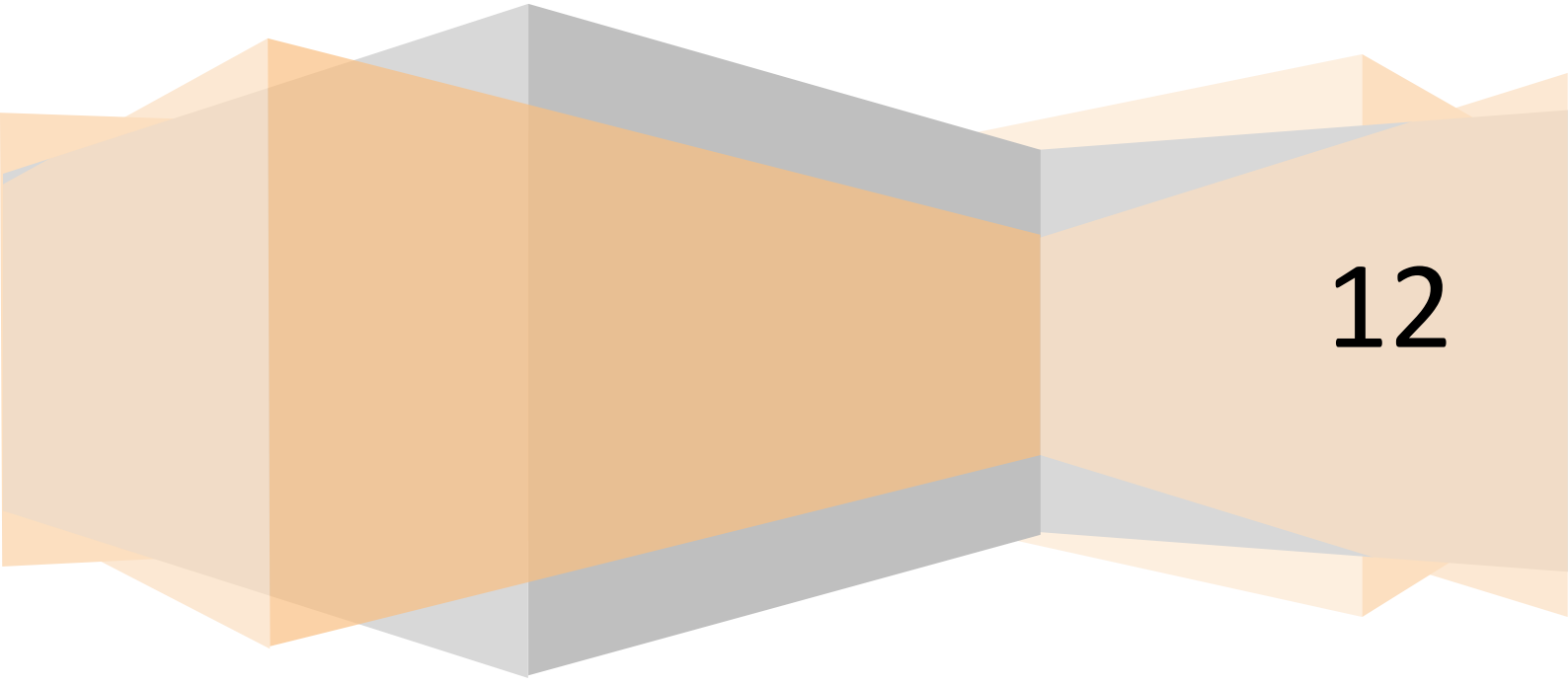
Grupp26

Architecture specification 0.1

Cloudio

Marcus Parkkinen



12

Table of contents

1 Introduction

1.1 Purpose of this document

This design document aims to provide support for new developers in understanding the purpose and role of the core components in the application. It outlines application module structure as well as intramodular and intermodular communication flow.

1.2 Document scope

The scope of this document is limited to classes that participate in the communication flow between the different modules. It does not mention resource classes or adapter classes for view module components.

1.3 Terminology

TBA.

2 Architecture overview

2.1 Application block diagram

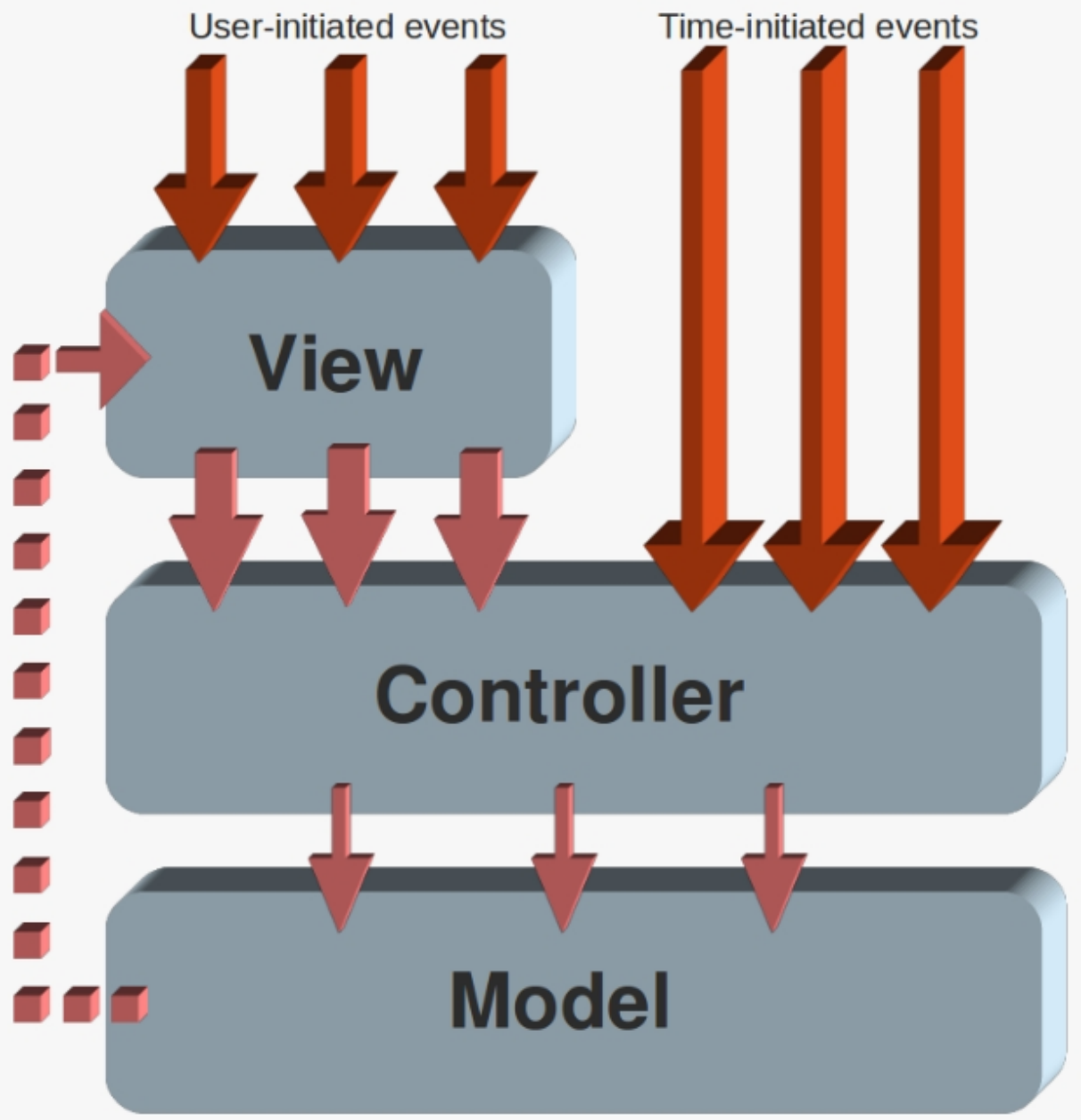


Figure 1 - Architecture block diagram. Darker arrows denote root sources for events that enter the application while brighter hue arrows depict information flow between application components.

The two main sources that trigger change within the application are user-initiated events and time-initiated events, as depicted in figure 1. User-initiated events constitute all forms of interaction with user controls contained in the view module such as buttons, seek bars and lists, while time-initiated events are triggered by threads contained in different parts of the controller module. In its original state the application is only susceptible to user-initiated events until the

user selects a book to play. User-initiated events are discussed in more detail in the following part.

2.2 View module

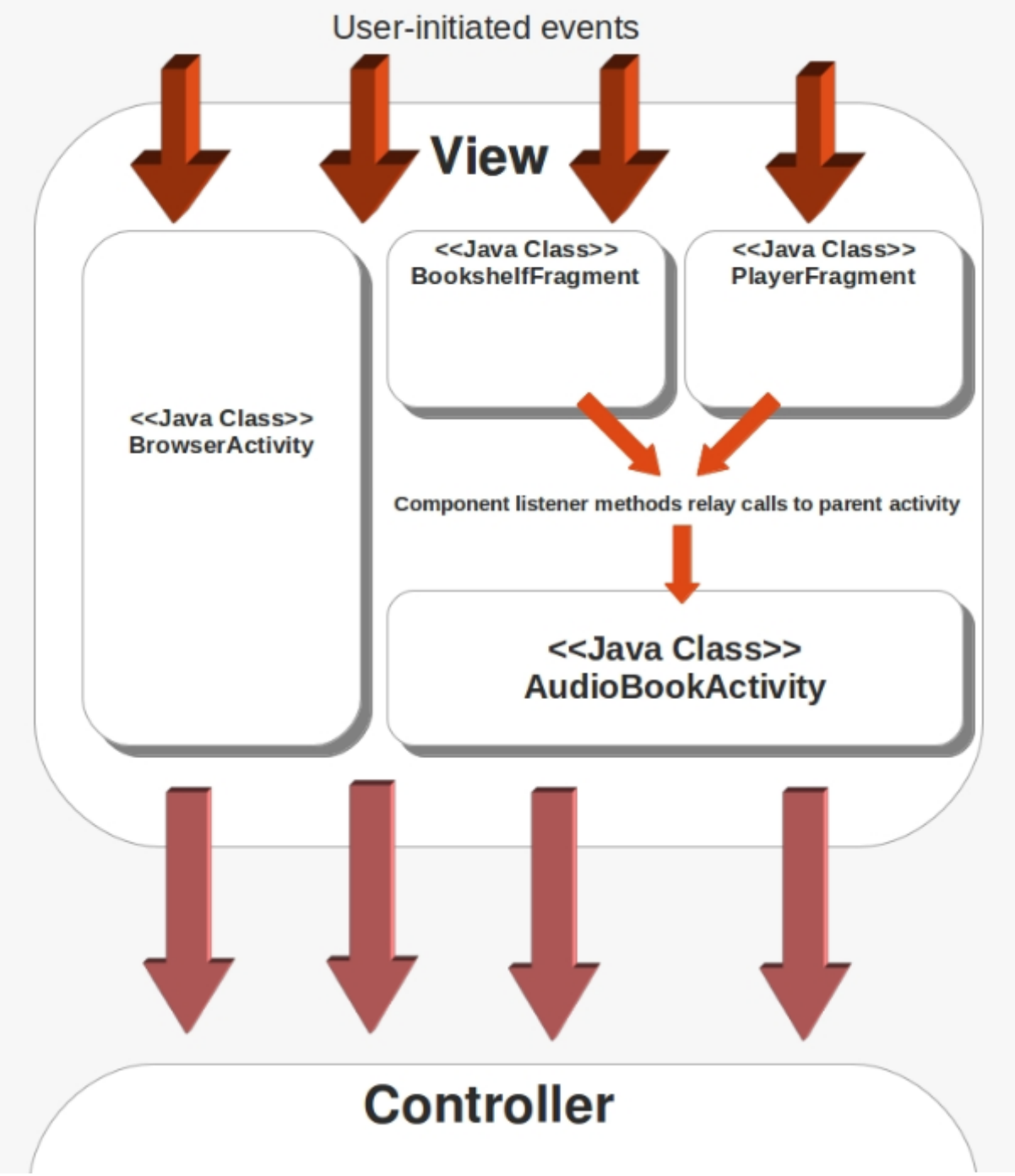


Figure 1.1 - Event information flow chart for the view module of the application.

The application contains three classes that provide elements that the user can interact with. As depicted in figure 1.1 above, these are *BrowserActivity*, *BookshelfFragment* and *PlayerFragment*. Depending on which of these component classes the user interacts with, events can either be relayed through a parent fragment container (*AudioBookActivity*) as in the case with the fragment classes, or be directly translated into a method call in the controller module as with the

BrowserActivity class. This way, each user-initiated interaction with a graphical component is indirectly mapped to a method call in a controller class, as depicted in figure 1.2.

2.3 Controller module

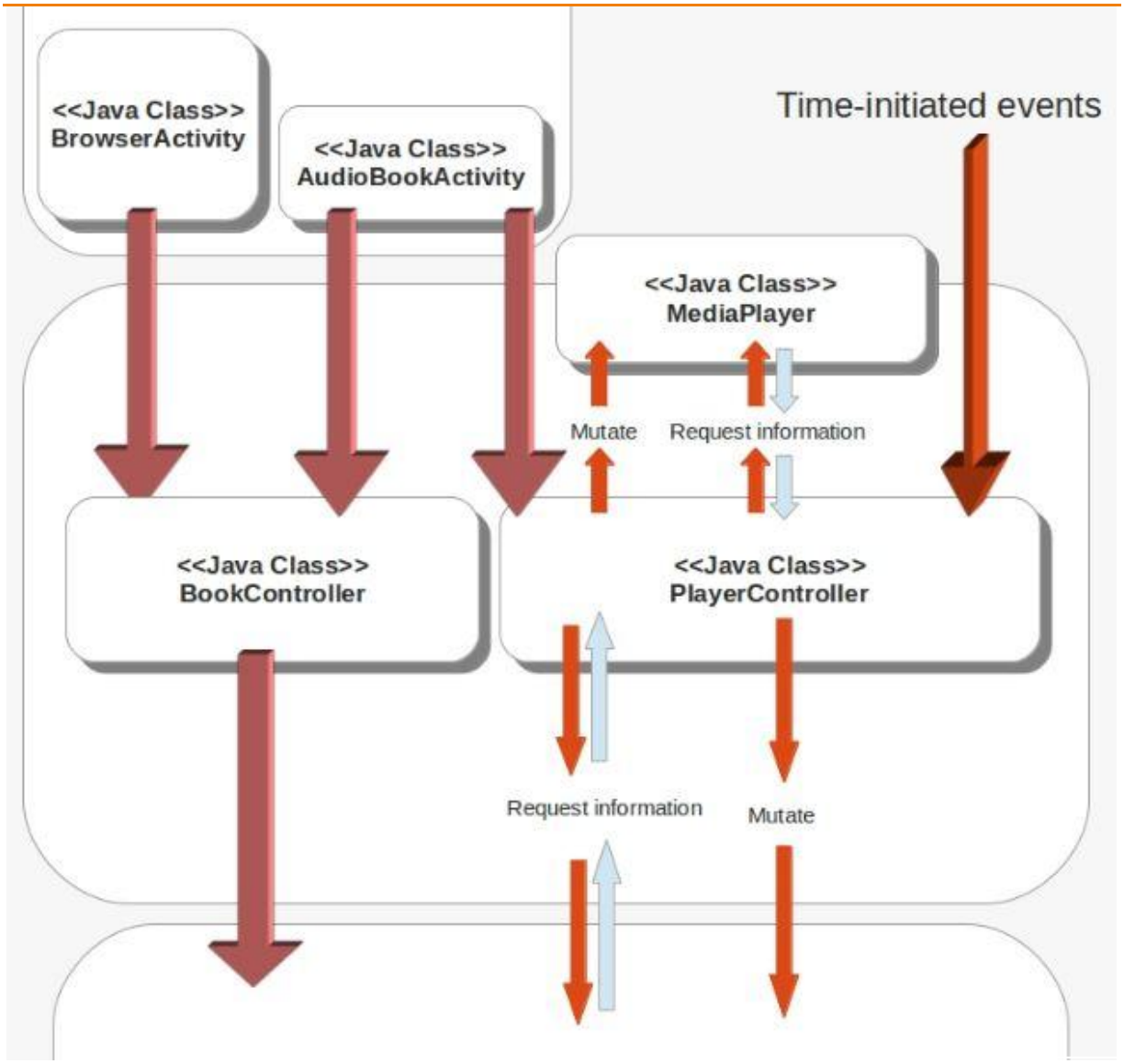


Figure 1.2 - Event information flow chart for the controller module of the application.

The main purpose of classes contained in the controller module is to... more TBA, until then: Lorem ipsum.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse diam mauris, pharetra at sollicitudin quis, aliquet eget nibh. Pellentesque pretium feugiat lectus nec interdum. Sed enim diam, interdum a sollicitudin id, facilisis eu libero. Aenean posuere adipiscing enim. Pellentesque sit amet sem sem, ultricies feugiat neque. In a tellus eros. Nunc ut dictum urna. Proin aliquam tortor erat. Etiam non lorem id leo tincidunt condimentum. Suspendisse mattis mauris faucibus

2.4 Model module

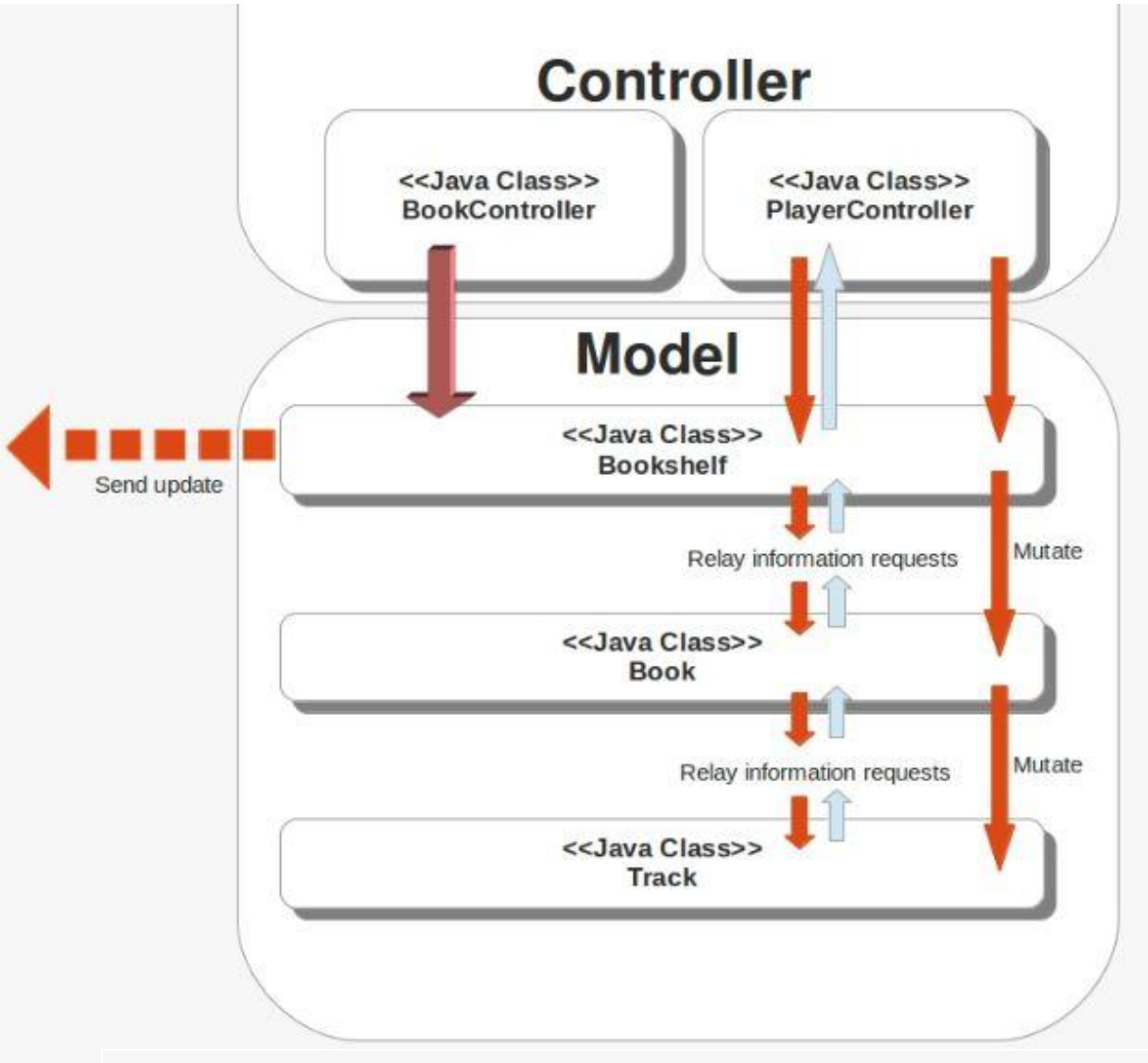


Figure 1.3 - Information flow chart for core classes in the model module of the application.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse diam mauris, pharetra at sollicitudin quis, aliquet eget nibh. Pellentesque pretium feugiat lectus nec interdum. Sed enim diam, interdum a sollicitudin id, facilisis eu libero. Aenean posuere adipiscing enim. Pellentesque sit amet sem sem, ultricies feugiat neque. In a tellus eros. Nunc ut dictum urna. Proin aliquam tortor erat. Etiam non lorem id leo tincidunt condimentum. Suspendisse mattis mauris faucibus

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse diam mauris, pharetra at sollicitudin quis, aliquet eget nibh. Pellentesque pretium feugiat lectus nec interdum. Sed enim diam, interdum a sollicitudin id, facilisis eu libero. Aenean posuere adipiscing enim. Pellentesque sit amet sem sem, ultricies feugiat neque. In a tellus eros. Nunc ut dictum urna. Proin aliquam tortor erat. Etiam non lorem id leo tincidunt condimentum. Suspendisse mattis mauris faucibus.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse diam mauris, pharetra at sollicitudin quis, aliquet eget nibh. Pellentesque pretium feugiat lectus nec interdum. Sed enim diam, interdum a sollicitudin id, facilisis eu libero. Aenean posuere adipiscing enim. Pellentesque sit amet sem sem, ultricies feugiat neque. In a tellus eros. Nunc ut dictum urna. Proin aliquam tortor erat. Etiam non lorem id leo tincidunt condimentum. Suspendisse mattis mauris faucibus

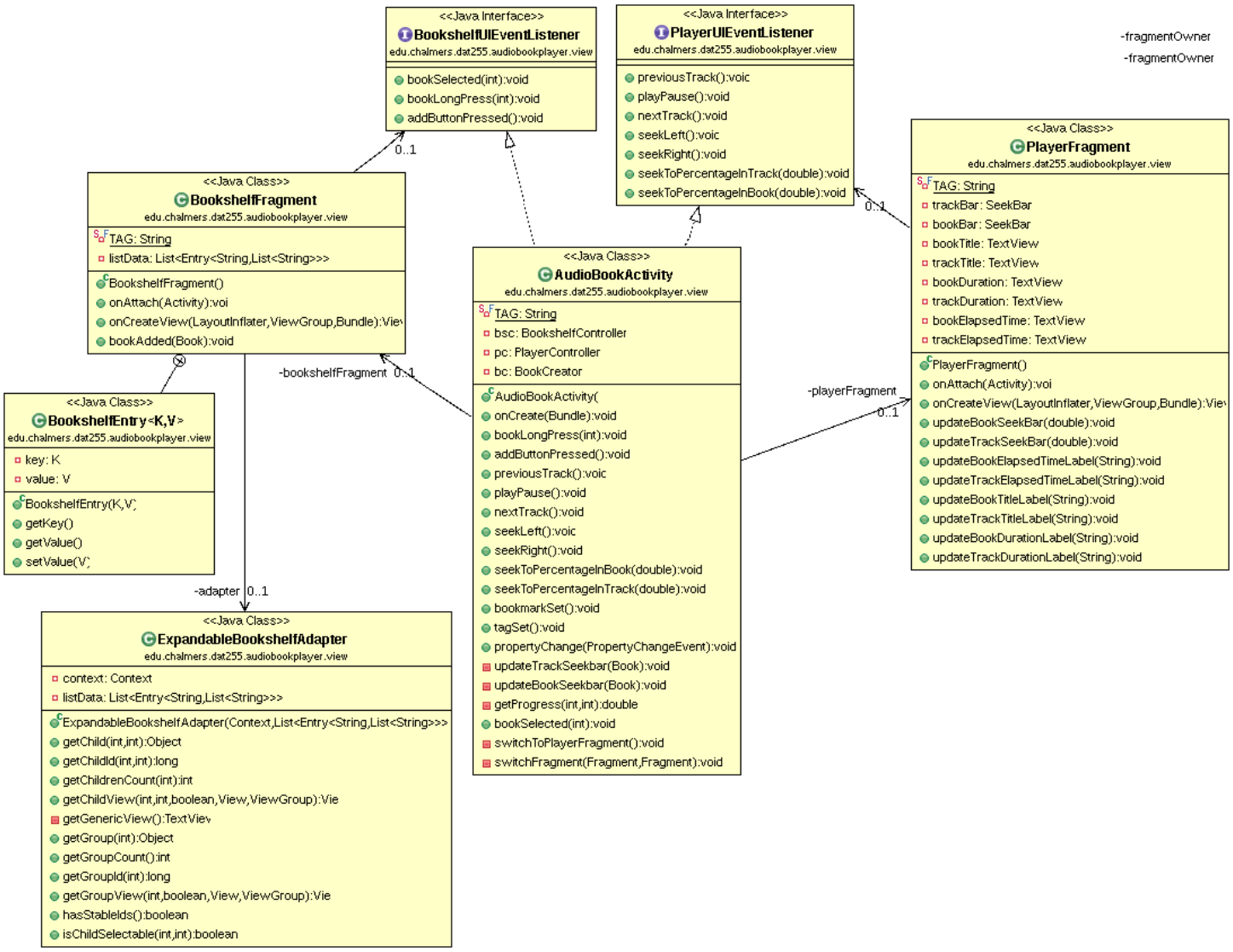
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse diam mauris, pharetra at sollicitudin quis, aliquet eget nibh. Pellentesque pretium feugiat lectus nec interdum. Sed enim diam, interdum a sollicitudin id, facilisis eu libero. Aenean posuere adipiscing enim. Pellentesque sit amet sem sem, ultricies feugiat neque. In a tellus eros. Nunc ut dictum urna. Proin aliquam tortor erat. Etiam non lorem id leo tincidunt condimentum. Suspendisse mattis mauris faucibus

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse diam mauris, pharetra at sollicitudin quis, aliquet eget nibh. Pellentesque pretium feugiat lectus nec interdum. Sed enim diam, interdum a sollicitudin id, facilisis eu libero. Aenean posuere adipiscing enim. Pellentesque sit amet sem sem, ultricies feugiat neque. In a tellus eros. Nunc ut dictum urna. Proin aliquam tortor erat. Etiam non lorem id leo tincidunt condimentum. Suspendisse mattis mauris faucibus

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse diam mauris, pharetra at sollicitudin quis, aliquet eget nibh. Pellentesque pretium feugiat lectus nec interdum. Sed enim diam, interdum a sollicitudin id, facilisis eu libero. Aenean posuere adipiscing enim. Pellentesque sit amet sem sem, ultricies feugiat neque. In a tellus eros. Nunc ut dictum urna. Proin aliquam tortor erat. Etiam non lorem id leo tincidunt condimentum. Suspendisse mattis mauris faucibus.

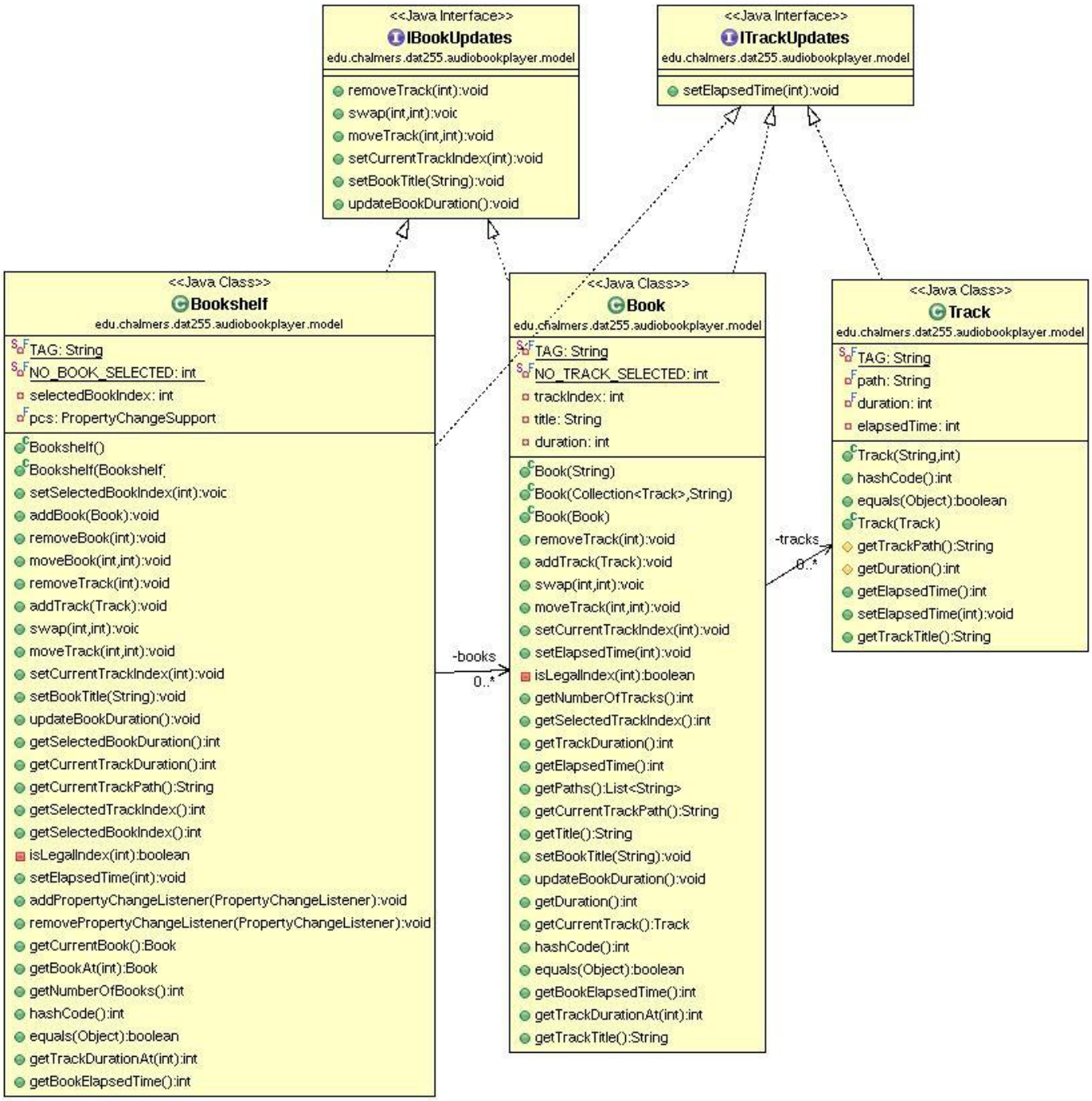
3.0 UML-diagrams

3.1 View module UML



3.2 Controller module UML

3.3 Model module UML



3.4 Complete application UML

See appendix 1.