

CS Department Automated Information Timeline
Assignment 6.3: GRASP Concepts

Matthew Hays, Pawan Bhandari, Sarah Faron, Tim Klimpel

February 19, 2024

Contents

| | | |
|----------|------------------------------------|----------|
| 1 | Introduction | 3 |
| 1.1 | Purpose | 3 |
| 2 | GRASP concepts | 3 |
| 3 | Annotated Sequence diagrams | 3 |
| 3.1 | Create Event | 4 |
| 3.2 | Review for Approval | 5 |
| 3.3 | Edit Post | 6 |
| 3.4 | Manage Displayed Media | 7 |
| 3.5 | Add Page to Event | 8 |

List of Figures

| | | |
|---|--|---|
| 1 | Sequence Diagram: Create Event | 4 |
| 2 | Sequence Diagram: Review for Approval | 5 |
| 3 | Sequence Diagram: Edit Post | 6 |
| 4 | Sequence Diagram: Manage Displayed Media | 7 |
| 5 | Sequence Diagram: Add page to Event | 8 |

1 Introduction

1.1 Purpose

The purpose of this assignment is to collaborate as a team to identify General Responsibility Assignment Software Patterns (GRASP) in the design of the CS Department Automated Information Timeline project. The team (aka The Incredibles) met multiple times over the course of a few days to work together and identify the GRASP concepts in the design. Section 2 of this document consists of the identified GRASP concepts and section 3 contains the annotated sequence diagrams.

2 GRASP concepts

Table below categorizes all the classes that appear on the sequence diagrams based on the GRASP concept they implement.

| Controller | Creator | Information Expert | Low Coupling/High Cohesion |
|--|---|--|--|
| EventController ReviewController PostController MediaController | EventMapper EventRepository NotificationRepository NotificationMapper PostRepository MediaRepository PageRepository | EventRepository NotificationRepository PostRepository MediaRepository PageRepository | EventService NotificationService ReviewService PostService MediaService DisplayService PageService |

Table 1: Classes with the identified GRASP concepts

Low coupling and high cohesion is achieved by clearly separating out Controller, Service and Repository actions so that each class is only responsible for its specific layer of the application. This is also a core part of the MVC pattern which is known to be loosely coupled and highly cohesive.

The mapper classes (EventMapper and NotificationMapper) were also identified as a creators because they are meant to be injected in service classes to create objects to be used as entities. Mapper classes support low coupling by removing a need for mapping user inputs directly to domain object needs and high cohesion by dealing only with JSON Serialization/deserialization of Event/EventDTO objects.

3 Annotated Sequence diagrams

A subset of sequence diagrams are annotated with the identified GRASP concepts and are included below. These sequence diagrams cover all the classes and the GRASP patterns they implement.

3.1 Create Event

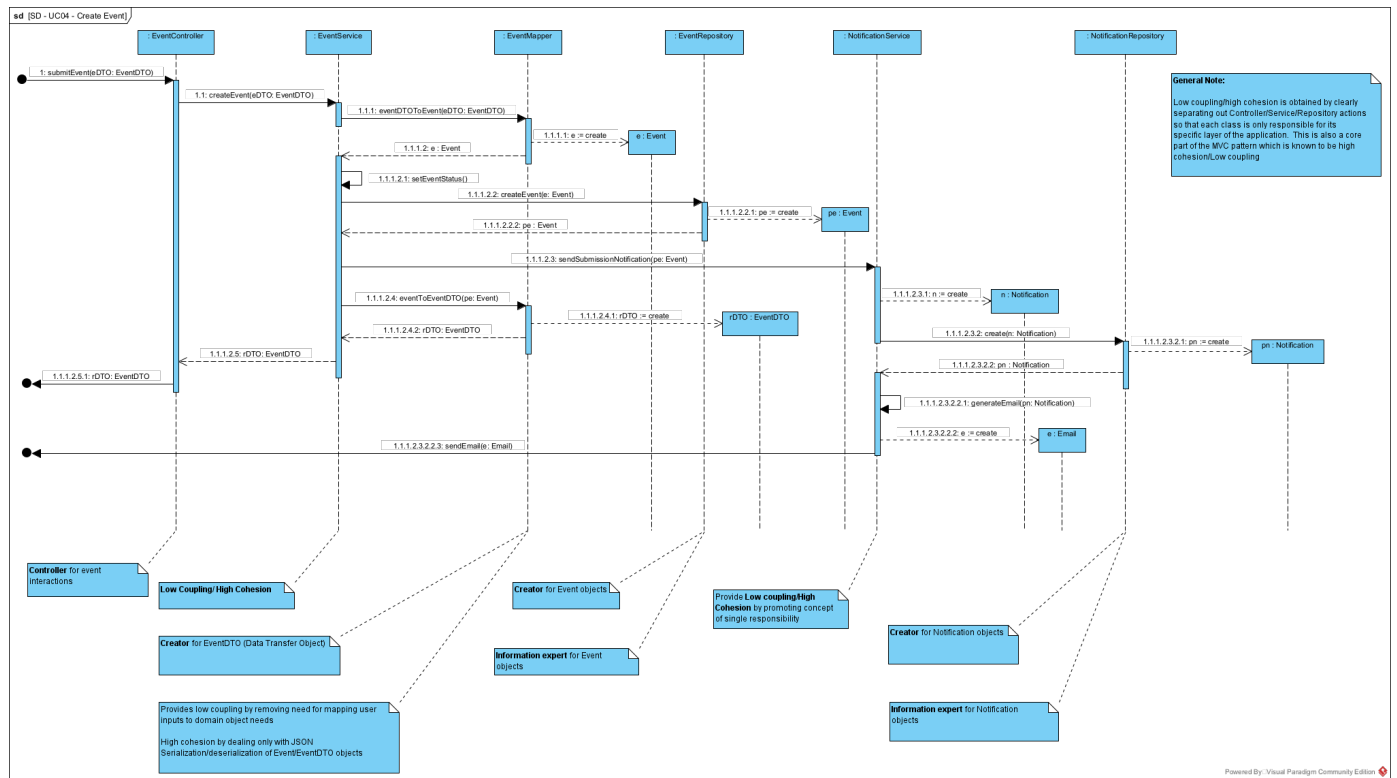


Figure 1: Sequence Diagram: Create Event

3.2 Review for Approval

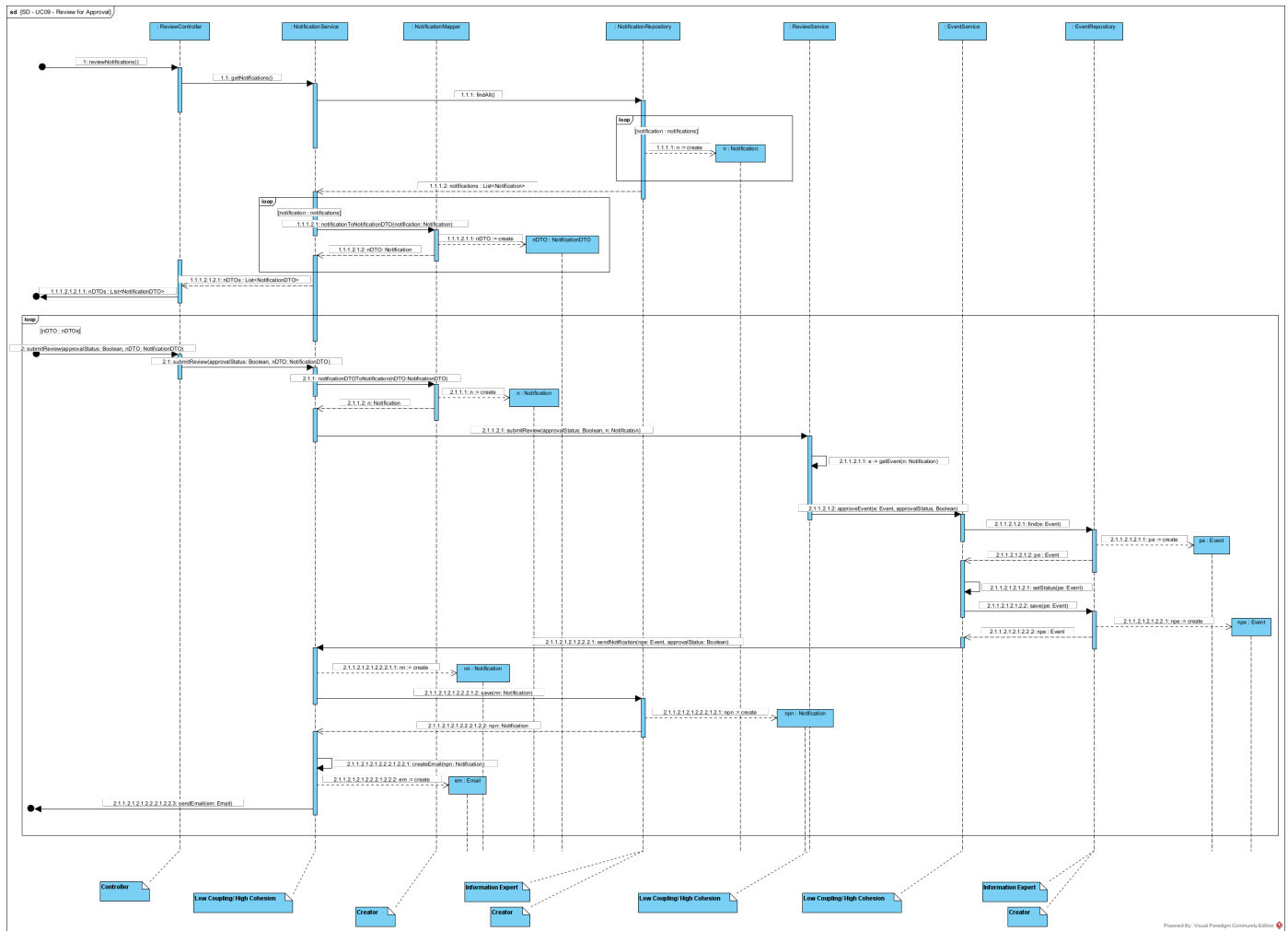


Figure 2: Sequence Diagram: Review for Approval

3.3 Edit Post

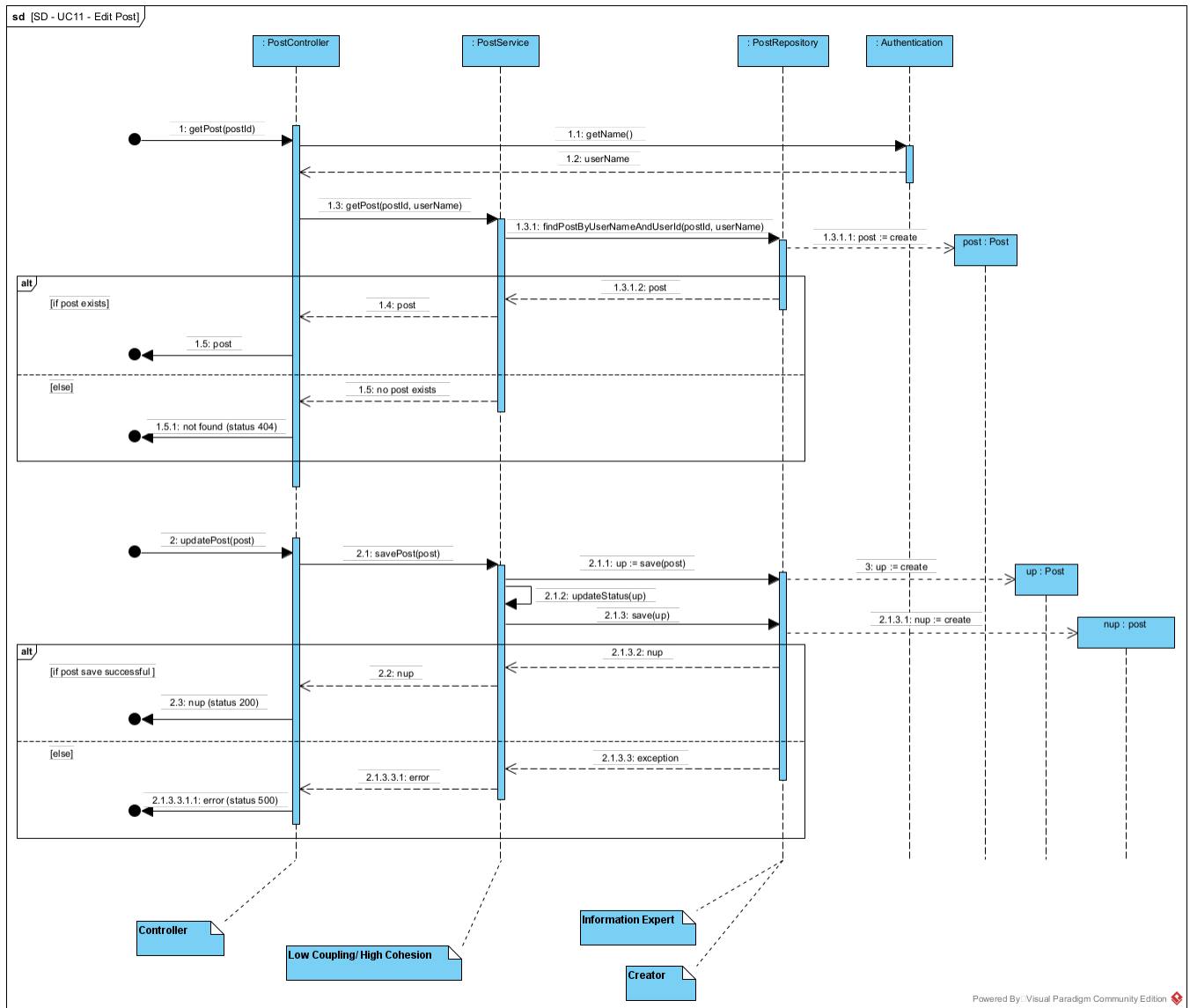


Figure 3: Sequence Diagram: Edit Post

3.4 Manage Displayed Media

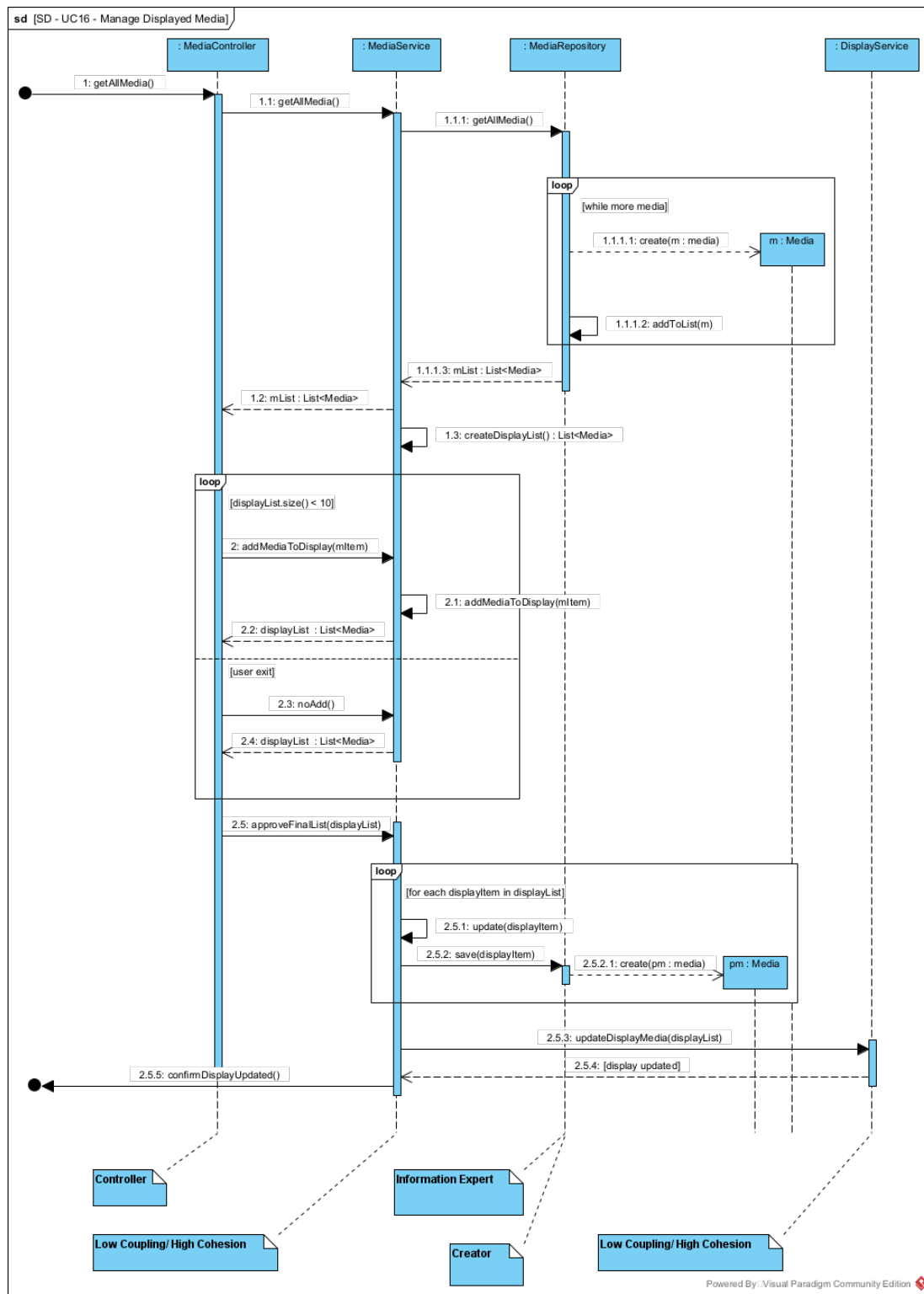


Figure 4: Sequence Diagram: Manage Displayed Media

3.5 Add Page to Event

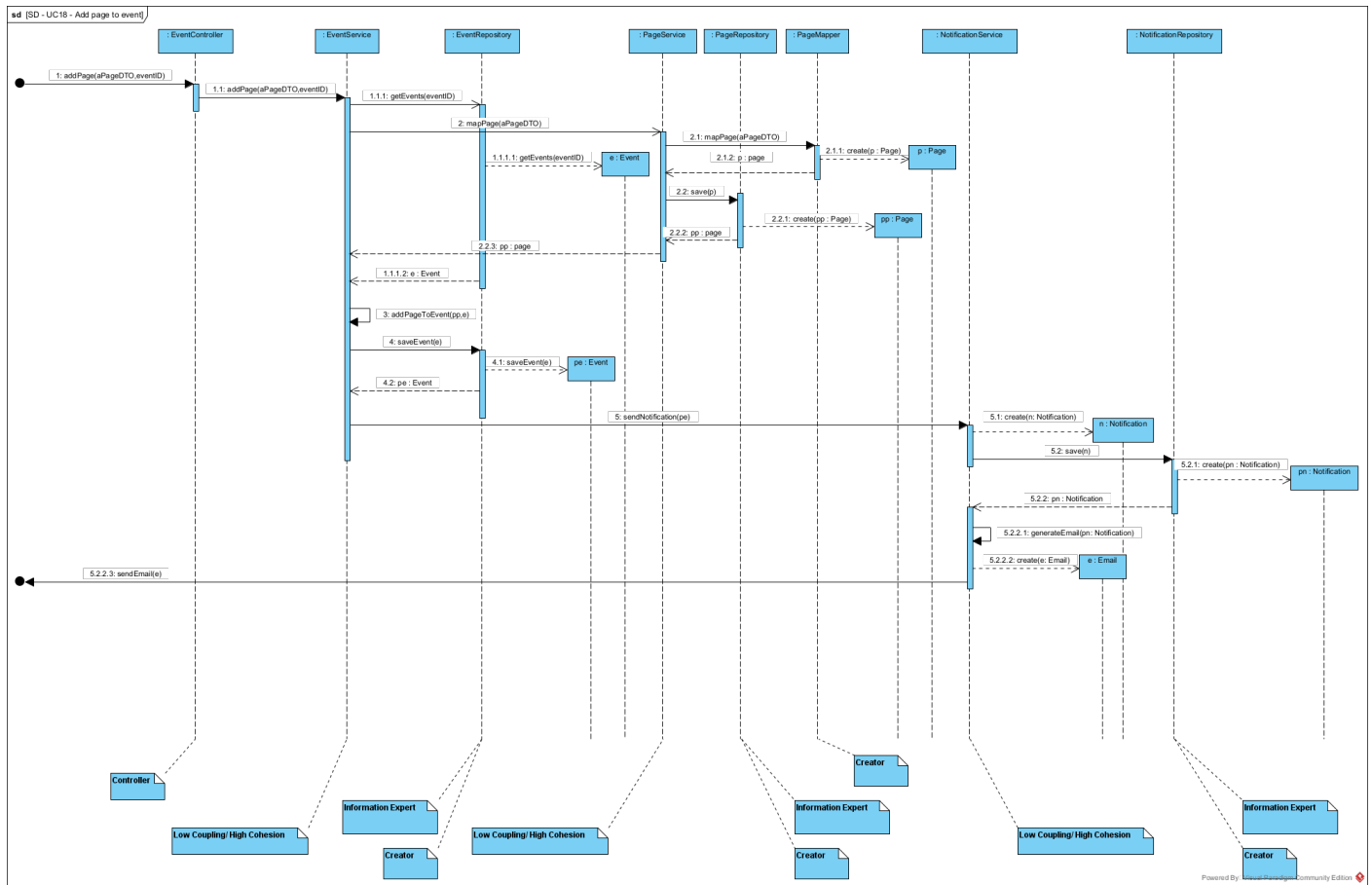


Figure 5: Sequence Diagram: Add page to Event