

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

SOFTWARE REQUIREMENTS SPECIFICATION AND
TECHNOLOGY NEUTRAL PROCESS DESIGN
(UNIVERSITY OF PRETORIA - POSTGRADUATE PAPER
INTERACTION PORTAL)

TEAM JULIET

QUINTON WEENINK
U13176545

VUYANI SHABANGU
U11171139

RUAN KLINKERT
U14022282

REINHARDT CROMHOUT
U14009936

ROHAN CHHIPA
U14188377

BRANDON WARDLEY
U29005150

Contents

1	Introduction	1
2	Vision	1
3	Background	2
4	Functional requirements and application design	2
4.1	Operator subsystem	2
4.1.1	Use cases	2
4.1.2	Services Contracts	3
4.1.3	Required functionality	6
4.1.4	Process specifications	9
4.2	Domain Model	12
5	Open Issues	13

1 Introduction

The purpose of this document is to describe the requirements of the Postgraduate Paper Interaction Portal application, by illustrating the purpose and uses of the system, as well as the development of the system. The system constraints, interface and various interactions will also be discussed, and testable quality requirements will be provided where applicable.

2 Vision

The Postgraduate Paper Interaction Portal is an application which allows the effective management of research groups, research projects, and researchers themselves, using either a web interface, or an android application. The application will allow the storage of metadata pertaining to papers which are being written or have been written, the authors who are involved with these papers, as well as the various publications to which these papers will be submitted. The application is intended for internal use by the University of Pretorias Department of Computer Science, and should be as simple and lightweight as possible.

A system administrator will be in charge of managing the system, and as such they can add and remove users, as well as change metadata when and where necessary. The administrator will also ensure that the system is properly audited, with all changes being recorded by the system itself.

The Head of Department (HoD) can view the status of all papers within the department, as well as their metadata, such that the progress and status of these papers can be tracked. The HoD can also view the statistics of each department, showing the number of researchers within that department, the number of publishing units that department has accrued, the number of papers they have released, and what the various performance goals of that department are.

Research leaders are in charge of all the researchers within their specific research group. They can view the status and progress of all papers to be published under their group, as well as the metadata associated with each paper. Research leaders can view the details of each researcher within their group, as well as add new researchers to their group as needed.

A normal user within the system is a researcher who falls under the authority of a specific research leader. The user can view the metadata of each paper of which they are an author or co-author, and are also able to change these details as necessary by editing stored data, changing the progress report of the paper, or changing the status of the paper. The user can also create new papers, provided that they are either an author or co-author of said paper.

Each user within the system is responsible for entering the required metadata for each user, researcher, paper or publication goal they introduce into the system. The system will store each new entry, such that it might quickly be retrieved for future use, thus saving the users time when entering data. Papers may not be deleted from the system, and each entry or change within the system must be recorded.

3 Background

The need for the Postgraduate Paper Interaction Portal became apparent after it was found that the system currently in use by the University of Pretoria's Department of Computer Science was no longer sufficient to effectively cater to the needs of the department. In its current state, this system consisted out of a single spreadsheet into which all the various details pertaining to research groups, projects, researchers and publications were inserted. These details were stored in a number of tables, and, using various mathematical formulae, a number of useful statistics were shown, usually pertaining to the number of publication units earned by researchers versus their goal. Whilst this system has the benefit of making all the stored information available at a single glance, the way in which this information is displayed is cluttered and confusing, a problem which becomes more apparent as the amount of stored information increases.

Research and the publication of findings is a crucial part of the department's work, and as such, a request was made for the development of an application which would allow the researchers and staff within the department to easily track and manage the details of their research projects. This functionality extends to the ability to easily add new projects, researchers and publishing goals into the system, with much of the management and processing of this information now being handled by the application itself, as opposed to the users.

If it is found that this system succeeds as an improvement over the old system, and that research personnel find that it provides a quality of life improvement when managing research projects, the application may be further improved and distributed such that it may be used by other departments within the University of Pretoria. Once it has been established that the application can be used by multiple departments, and that it notably simplifies and improves the research management process, the application may be marketed for use in other institutions.

4 Functional requirements and application design

In order to represent all of the use cases associated with the system we have divided the system up into sections that will indicate all of the use cases associated with each and how they include or expand upon other use cases.

4.1 Operator subsystem

4.1.1 Use cases

Critical

- Add drone
- Remove drone
- Assign research group leader
- View research group unit score

Nice-To-Have

- Edit drone

4.1.2 Services Contracts

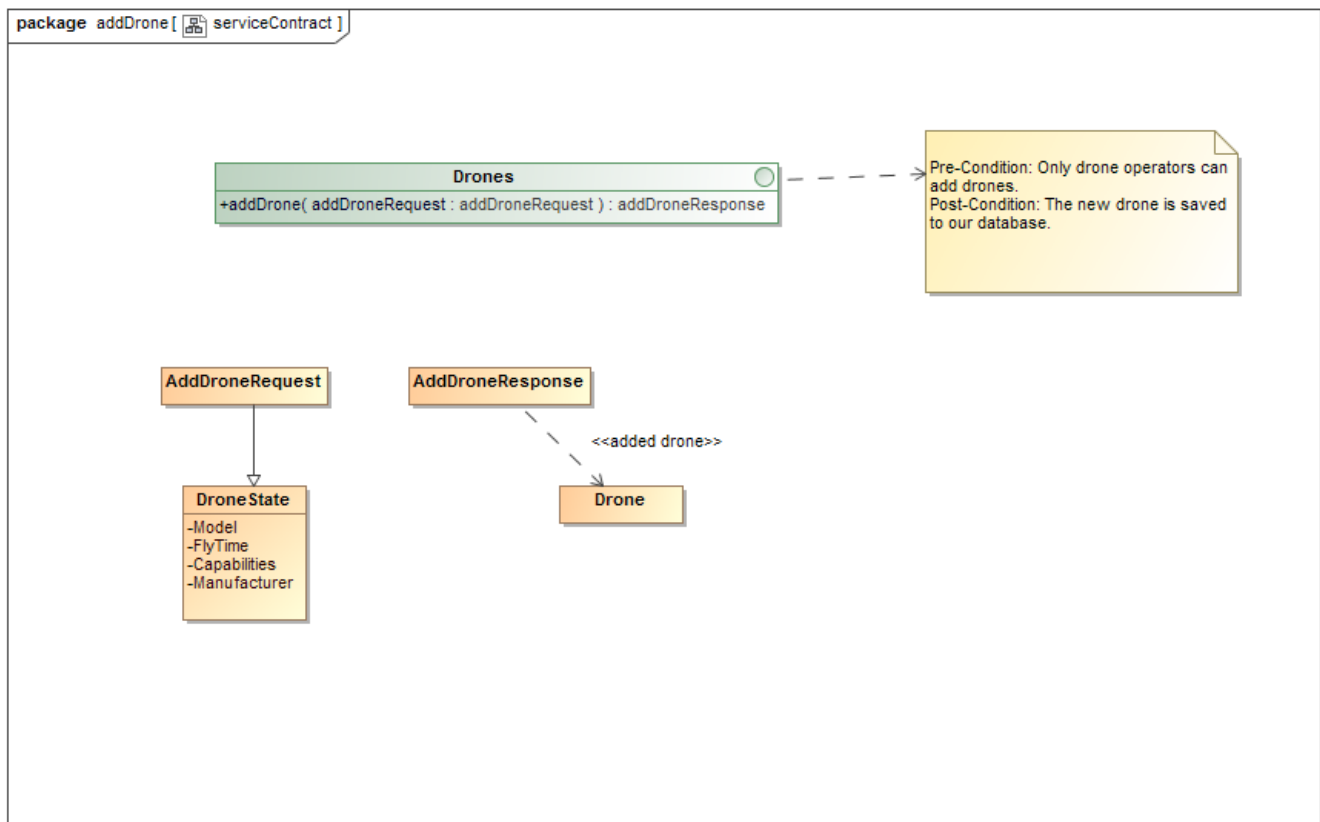


Figure 1: Services Contract : addDrone

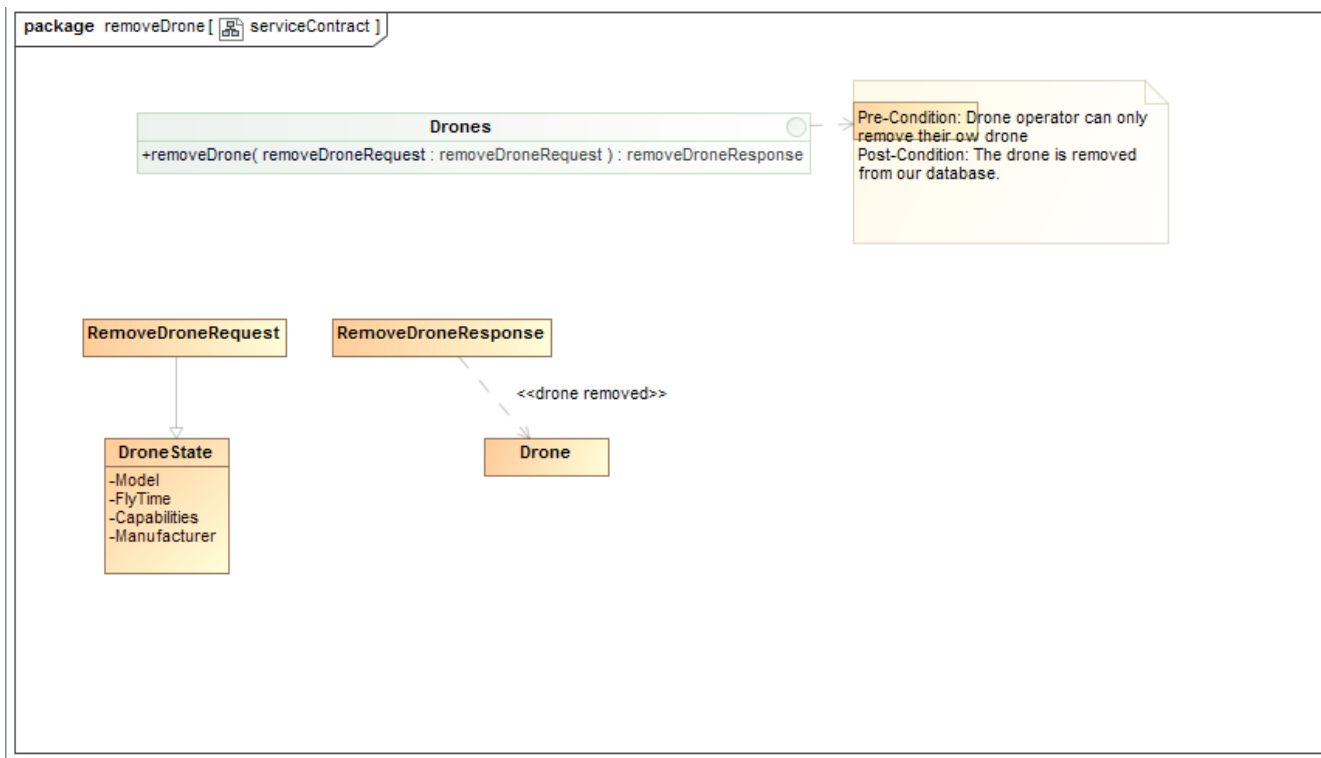


Figure 2: Services Contract : removeDrone

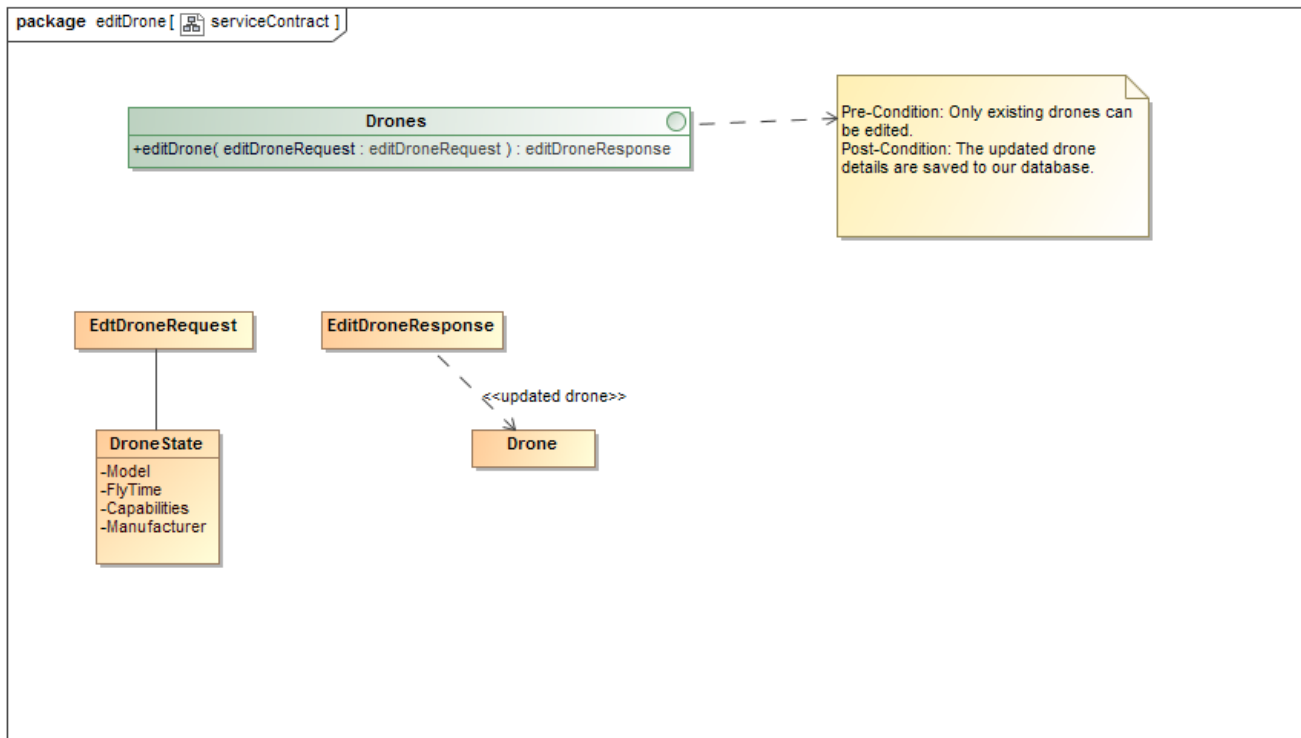


Figure 3: Services Contract : editDrone

4.1.3 Required functionality

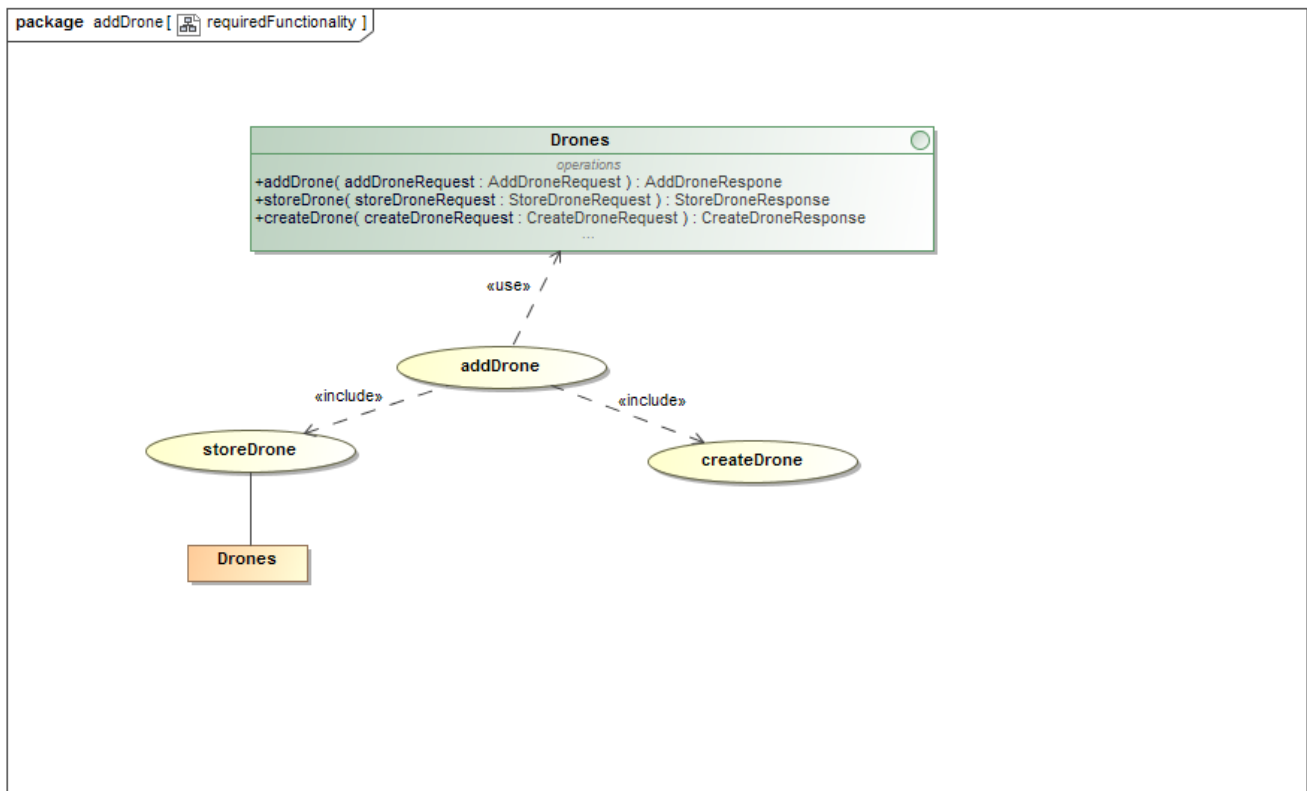


Figure 4: Functional Requirements : addDrone

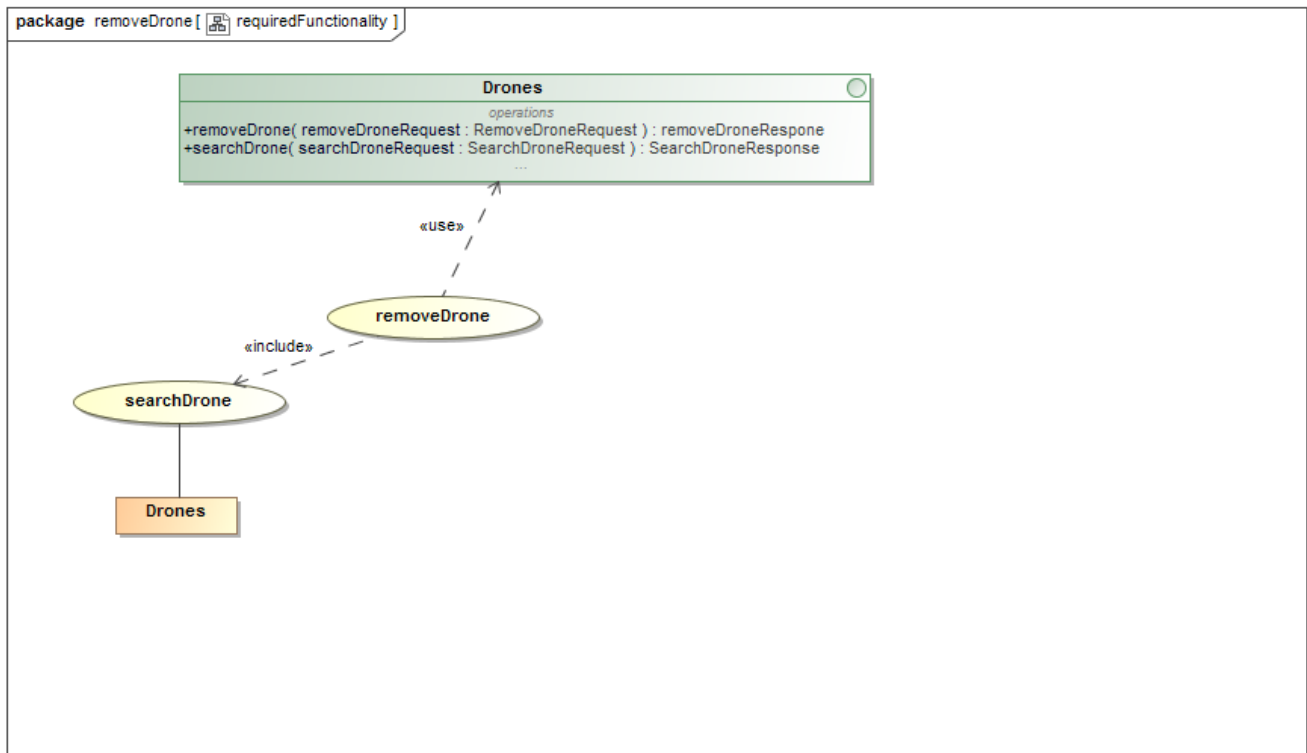


Figure 5: Functional Requirements : removeDrone

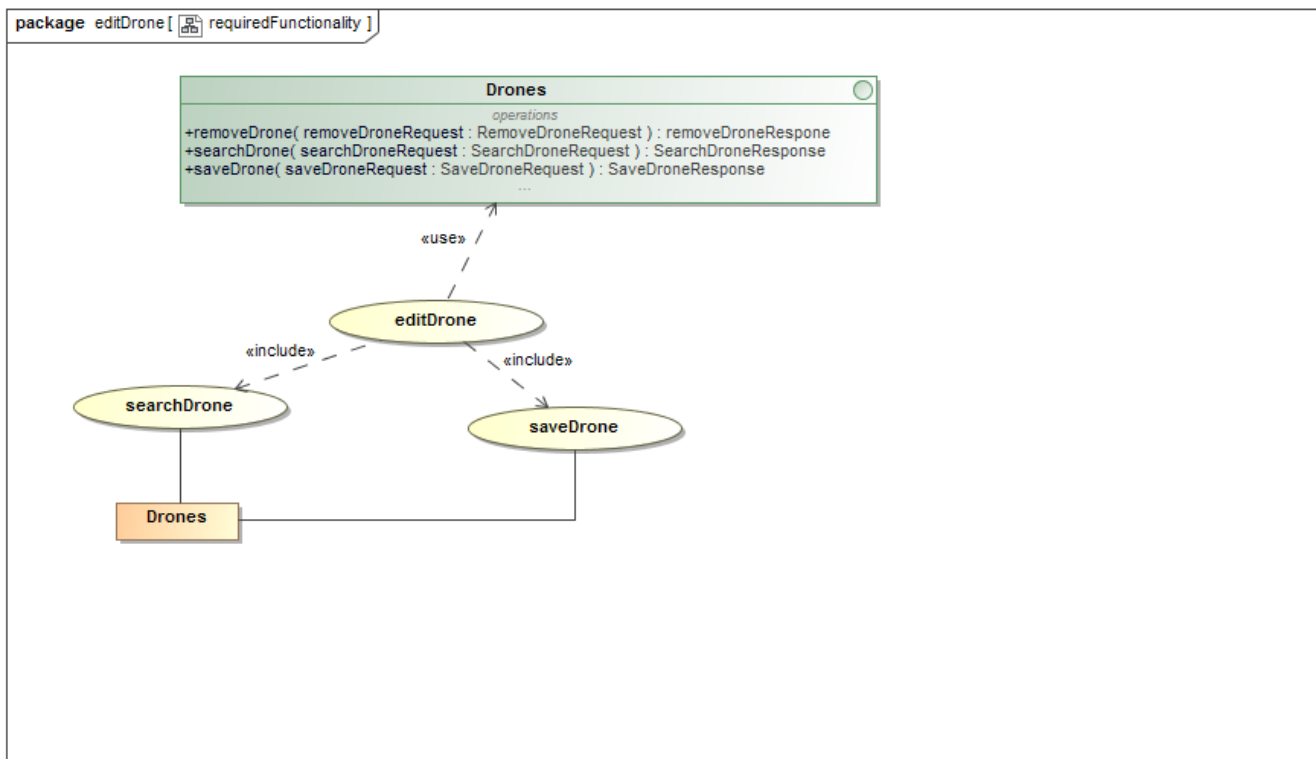


Figure 6: Functional Requirements : editDrone

4.1.4 Process specifications

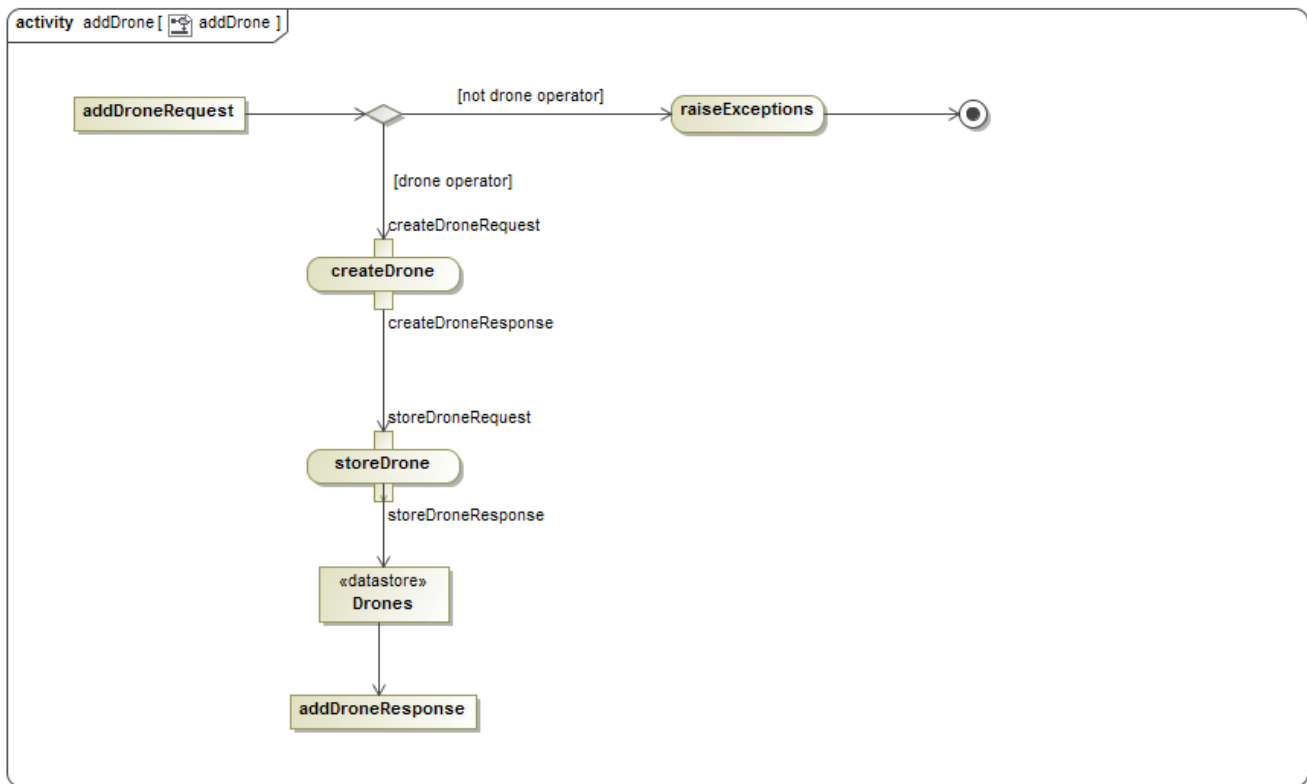


Figure 7: Process Specification : addDrone

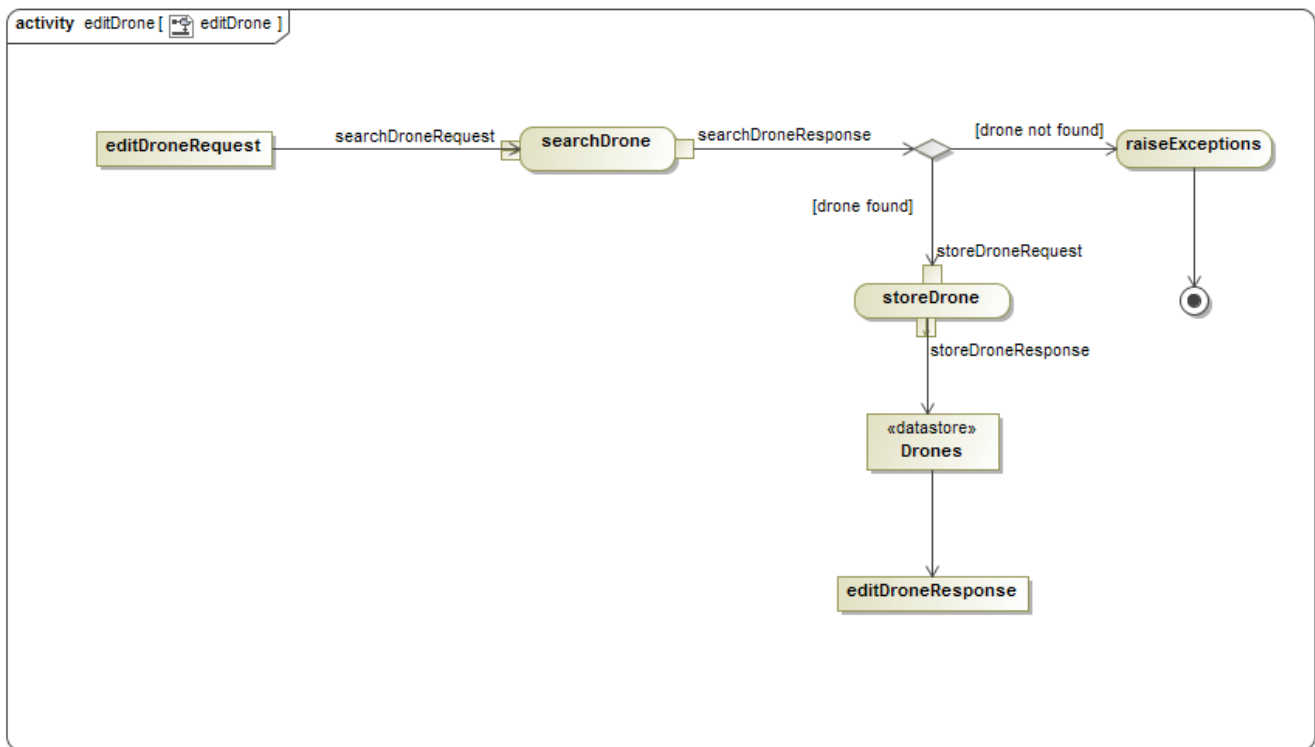


Figure 8: Process Specification : editDrone

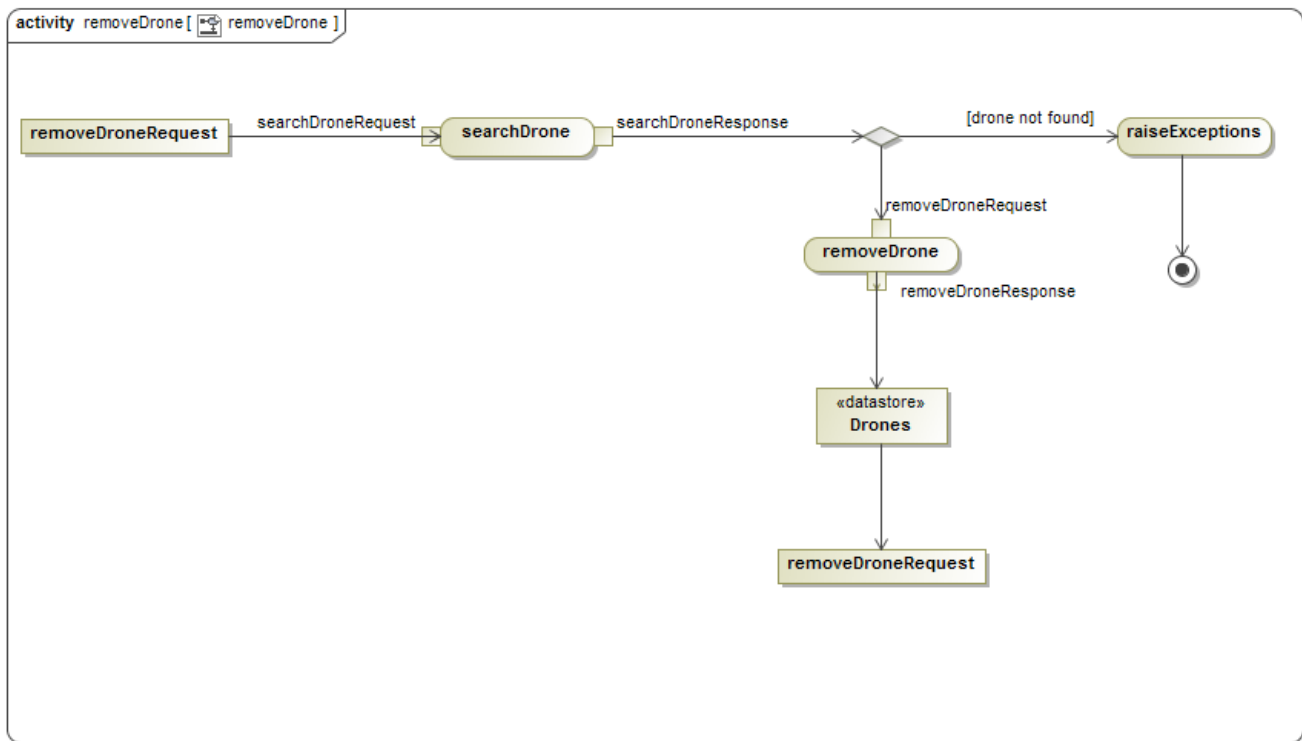


Figure 9: Process Specification : removeDrone

4.2 Domain Model

The domain model shows the data structure requirements of the Researcher Support System, as well as the relationships between the different domain objects.

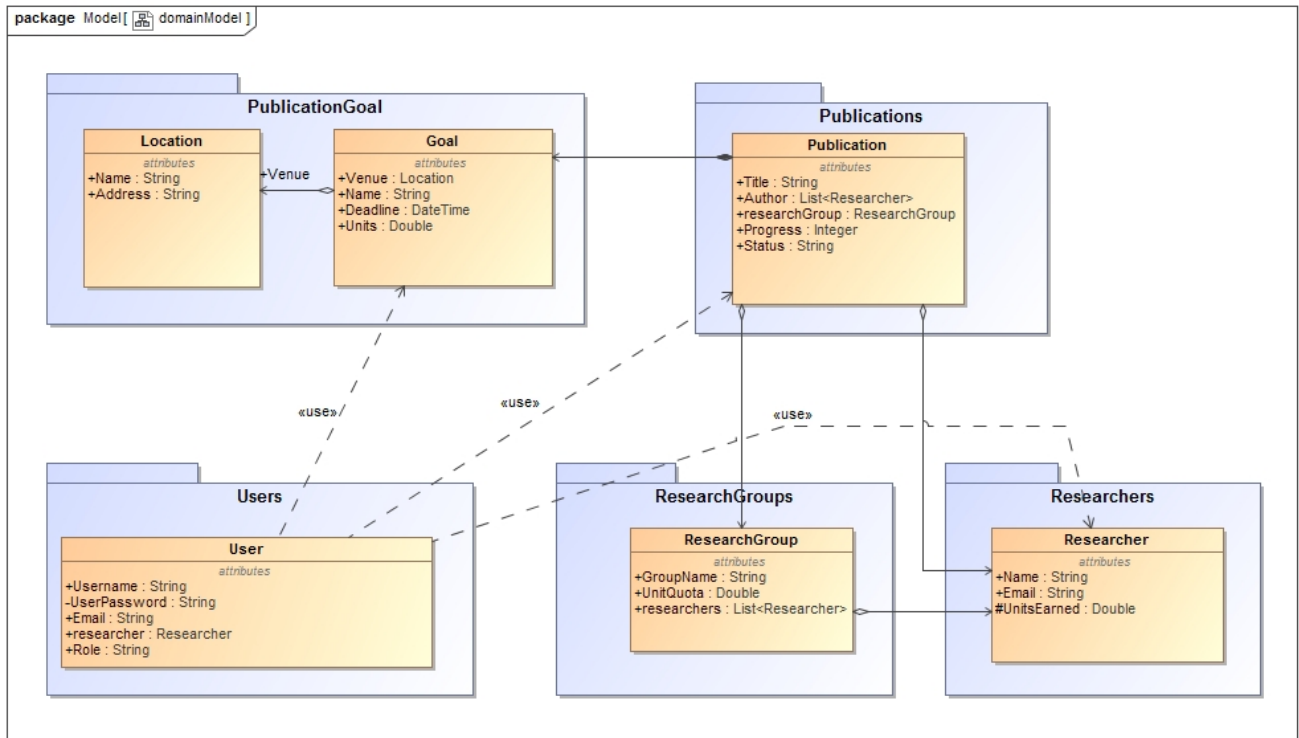


Figure 10: Domain Model

5 Open Issues

At the moment, there is uncertainty as to what resources will be made available by the client. These specific resources may determine the nature of the web server, and the approach used to implement the application.