

Module Interface Specification for Movie Recommender

Seyed Ali Mousavi

March 19, 2024

1 Revision History

Date	Version	Notes
2024-03-19	1.0	Initial Release

2 Symbols, Abbreviations and Acronyms

See SRS Documentation at

<https://github.com/alimousavi1997/RecommSys/blob/main/docs/SRS/SRS.pdf>

Contents

1	Revision History	i
2	Symbols, Abbreviations and Acronyms	ii
3	Introduction	1
4	Notation	1
5	Module Decomposition	1
6	MIS of KNN	3
6.1	Module	3
6.2	Uses	3
6.3	Syntax	3
6.3.1	Exported Constants	3
6.3.2	Exported Access Programs	3
6.4	Semantics	3
6.4.1	State Variables	3
6.4.2	Environment Variables	3
6.4.3	Assumptions	3
6.4.4	Access Routine Semantics	3
6.4.5	Local Functions	4
7	MIS of Item Selector	5
7.1	Module	5
7.2	Uses	5
7.3	Syntax	5
7.3.1	Exported Constants	5
7.3.2	Exported Access Programs	5
7.4	Semantics	5
7.4.1	State Variables	5
7.4.2	Environment Variables	5
7.4.3	Assumptions	5
7.4.4	Access Routine Semantics	5
7.4.5	Local Functions	6
8	Appendix	7
9	Reflection	7

3 Introduction

The following document details the Module Interface Specifications for Fill in your project name and description

Complementary documents include the System Requirement Specifications and Module Guide. The full documentation and implementation can be found at <https://github.com/alimousavi1997/RecommSys>.

4 Notation

You should describe your notation. You can use what is below as a starting point.

The structure of the MIS for modules comes from [?](#), with the addition that template modules have been adapted from [?](#). The mathematical notation comes from Chapter 3 of [?](#). For instance, the symbol $:=$ is used for a multiple assignment statement and conditional rules follow the form $(c_1 \Rightarrow r_1 | c_2 \Rightarrow r_2 | \dots | c_n \Rightarrow r_n)$.

The following table summarizes the primitive data types used by [?](#).

Data Type	Notation	Description
character	char	a single symbol or digit
integer	\mathbb{Z}	a number without a fractional component in $(-\infty, \infty)$
natural number	\mathbb{N}	a number without a fractional component in $[1, \infty)$
real	\mathbb{R}	any number in $(-\infty, \infty)$

The specification of [?](#) uses some derived data types: sequences, strings, and tuples. Sequences are lists filled with elements of the same data type. Strings are sequences of characters. Tuples contain a list of values, potentially of different types. In addition, [?](#) uses functions, which are defined by the data types of their inputs and outputs. Local functions are described by giving their type signature followed by their specification.

5 Module Decomposition

The following table is taken directly from the Module Guide document for this project.

Level 1	Level 2
Hardware-Hiding Module	
	?
	?
	?
Behaviour-Hiding Module	?
	Item Selector Module
Software Decision Module	KNN Module

Table 1: Module Hierarchy

6 MIS of KNN

6.1 Module

KNN

6.2 Uses

None

6.3 Syntax

6.3.1 Exported Constants

6.3.2 Exported Access Programs

Name	In	Out	Exceptions
accessProg	-	-	-

6.4 Semantics

6.4.1 State Variables

Not all modules will have state variables. State variables give the module a memory.

6.4.2 Environment Variables

This section is not necessary for all modules. Its purpose is to capture when the module has external interaction with the environment, such as for a device driver, screen interface, keyboard, file, etc.

6.4.3 Assumptions

Try to minimize assumptions and anticipate programmer errors via exceptions, but for practical purposes assumptions are sometimes appropriate.

6.4.4 Access Routine Semantics

accessProg():

- transition: if appropriate
- output: if appropriate
- exception: if appropriate

A module without environment variables or state variables is unlikely to have a state transition. In this case a state transition can only occur if the module is changing the state of another module.

Modules rarely have both a transition and an output. In most cases you will have one or the other.

6.4.5 Local Functions

As appropriate These functions are for the purpose of specification. They are not necessarily something that is going to be implemented explicitly. Even if they are implemented, they are not exported; they only have local scope.

7 MIS of Item Selector

7.1 Module

Itemselector

7.2 Uses

KNN

7.3 Syntax

7.3.1 Exported Constants

7.3.2 Exported Access Programs

Name	In	Out	Exceptions
accessProg	-	-	-

7.4 Semantics

7.4.1 State Variables

Not all modules will have state variables. State variables give the module a memory.

7.4.2 Environment Variables

This section is not necessary for all modules. Its purpose is to capture when the module has external interaction with the environment, such as for a device driver, screen interface, keyboard, file, etc.

7.4.3 Assumptions

Try to minimize assumptions and anticipate programmer errors via exceptions, but for practical purposes assumptions are sometimes appropriate.

7.4.4 Access Routine Semantics

accessProg():

- transition: if appropriate
- output: if appropriate
- exception: if appropriate

A module without environment variables or state variables is unlikely to have a state transition. In this case a state transition can only occur if the module is changing the state of another module.

Modules rarely have both a transition and an output. In most cases you will have one or the other.

7.4.5 Local Functions

As appropriate These functions are for the purpose of specification. They are not necessarily something that is going to be implemented explicitly. Even if they are implemented, they are not exported; they only have local scope.

References

8 Appendix

Extra information if required

9 Reflection

The information in this section will be used to evaluate the team members on the graduate attribute of Problem Analysis and Design. Please answer the following questions:

1. What are the limitations of your solution? Put another way, given unlimited resources, what could you do to make the project better? (LO_ProbSolutions)
2. Give a brief overview of other design solutions you considered. What are the benefits and tradeoffs of those other designs compared with the chosen design? From all the potential options, why did you select the documented design? (LO_Explores)