

Model

Package in package "

Model
Version Phase 1.0 Proposed
ASUS created on 30-Jan-23. Last modified 30-Jan-23

Use Case Model

Package in package 'Model'

Use Case Model
Version Phase 1.0 Proposed
muhiq created on 13-Nov-22. Last modified 22-Nov-22

Use Case Model diagram

Use Case diagram in package 'Use Case Model'

Use Case Model
Version 1.0
muhiq created on 13-Nov-22. Last modified 29-Jan-23

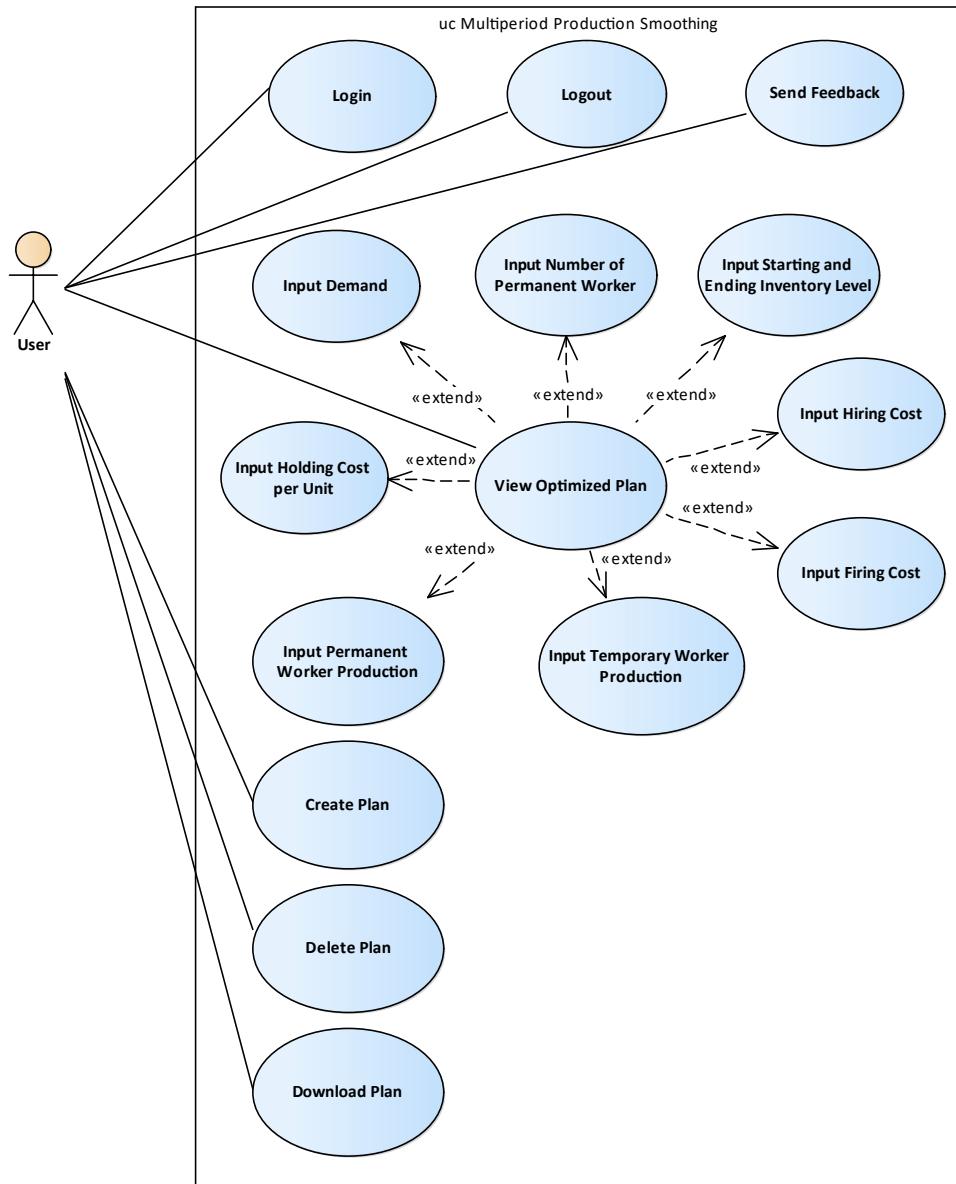


Figure 1: Use Case Model

uc Multiperiod Production Smoothing

Boundary in package 'Use Case Model'

uc Multiperiod Production Smoothing
 Version 1.0 Phase 1.0 Proposed
 muhiq created on 13-Nov-22. Last modified 29-Jan-23
 Extends

pulp

Package in package 'Use Case Model'

pulp
 Version 1.0 Phase 1.0 Proposed
 ASUS created on 29-Jan-23. Last modified 25-Mar-11

Flow Chart diagram

Flow Chart diagram in package 'pulp'

Flow Chart
Version 1.0

ASUS created on 29-Jan-23. Last modified 29-Jan-23

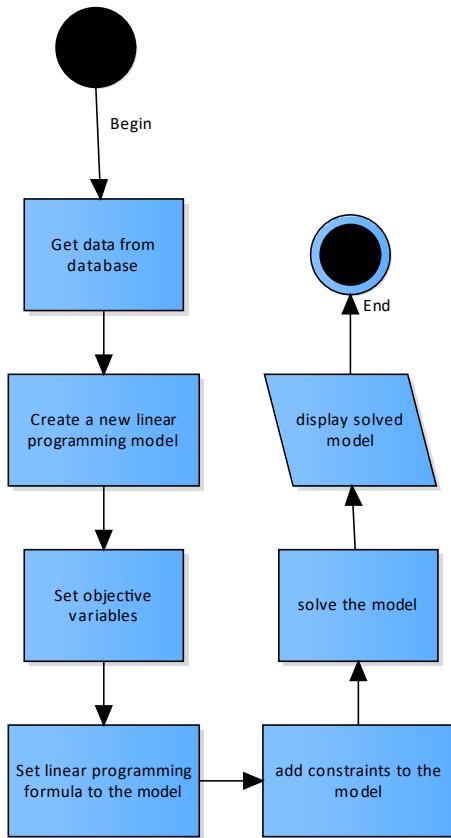


Figure 2: Flow Chart

Begin

ActivityInitial «FC_Begin» in package 'pulp'

End

ActivityFinal «FC_End» in package 'pulp'

Display1

Display «FC_Display» in package 'pulp'

Display1

Version 1.0 Phase 1.0 Proposed
ASUS created on 29-Jan-23. Last modified 29-Jan-23

display solved model

Process «FC_InputOutput» in package 'pulp'

display solved model
Version 1.0 Phase 1.0 Proposed
ASUS created on 29-Jan-23. Last modified 29-Jan-23

Create a new linear programming model

Process «FC_Process» in package 'pulp'

Create a new linear programming model
Version 1.0 Phase 1.0 Proposed
ASUS created on 29-Jan-23. Last modified 29-Jan-23

Get data from database

Process «FC_Process» in package 'pulp'

Get data from database
Version 1.0 Phase 1.0 Proposed
ASUS created on 29-Jan-23. Last modified 29-Jan-23

Process1

Process «FC_Process» in package 'pulp'

Process1
Version 1.0 Phase 1.0 Proposed
ASUS created on 29-Jan-23. Last modified 29-Jan-23

Set linear programming formula to the model

Process «FC_Process» in package 'pulp'

Set linear programming formula to the model
Version 1.0 Phase 1.0 Proposed
ASUS created on 29-Jan-23. Last modified 29-Jan-23

Set objective variables

Process «FC_Process» in package 'pulp'

Set objective variables
Version 1.0 Phase 1.0 Proposed
ASUS created on 29-Jan-23. Last modified 29-Jan-23

add constraints to the model

Process «FC_Process» in package 'pulp'

add constraints to the model
Version 1.0 Phase 1.0 Proposed
ASUS created on 29-Jan-23. Last modified 29-Jan-23

solve the model

Process «FC_Process» in package 'pulp'

solve the model
Version 1.0 Phase 1.0 Proposed
ASUS created on 29-Jan-23. Last modified 29-Jan-23

Begin

ActivityInitial «FC_Begin» in package 'pulp'

End

ActivityFinal «FC_End» in package 'pulp'

User

Actor in package 'Use Case Model'

User
Version 1.0 Phase 1.0 Proposed
ASUS created on 22-Nov-22. Last modified 29-Jan-23

CONNECTORS
<p>↗ UseCaseLink Source -> Destination From: User : Actor, Public To: Login : UseCase, Public</p>
<p>↗ UseCaseLink Source -> Destination From: User : Actor, Public To: Download Plan : UseCase, Public</p>
<p>↗ UseCaseLink Source -> Destination From: User : Actor, Public To: Send Feedback : UseCase, Public</p>
<p>↗ UseCaseLink Source -> Destination From: User : Actor, Public To: Logout : UseCase, Public</p>

CONNECTORS	
 UseCaseLink	Source -> Destination From: User : Actor, Public To: Delete Plan : UseCase, Public
 UseCaseLink	Source -> Destination From: User : Actor, Public To: Create Plan : UseCase, Public
 UseCaseLink	Source -> Destination From: User : Actor, Public To: View Optimized Plan : UseCase, Public

Create Plan

UseCase in package 'Use Case Model'

Create Plan
Version 1.0 Phase 1.0 Proposed
ASUS created on 29-Jan-23. Last modified 29-Jan-23

ELEMENTS OWNED BY Create Plan	
 Create Plan_ActivityGraph	: Activity
 Create Plan_SequenceDiagram	: Interaction

SCENARIOS	
 Basic Path.	Basic Path
1.	User input new plan's name
2.	User select new plan's month range
3.	User click on submit button
4.	System generate new database table into database
5.	System display available plans in database

CONSTRAINTS	
 Pre-condition.	User logged in to the system [Mandatory, Weight is 0.]
 Post-condition.	New plan is created [Approved, Weight is 1.]

CONNECTORS	
------------	--

CONNECTORS **UseCaseLink** Source -> DestinationFrom: User : Actor, Public
To: Create Plan : UseCase, Public**Create Plan_ActivityGraph***Activity owned by 'Create Plan', in package 'Use Case Model'*

Create Plan_ActivityGraph

Version 1.0 Phase 1.0 Proposed

ASUS created on 29-Jan-23. Last modified 29-Jan-23

ELEMENTS OWNED BY Create Plan_ActivityGraph System display available plans in database : Activity System generate new database table into database : Activity User click on submit button : Activity User input new plan's name : Activity User select new plan's month range : Activity End : ActivityFinal Start : ActivityInitial**Create Plan_ActivityGraph diagram***Activity diagram in package 'Use Case Model'*

Create Plan_ActivityGraph

Version 1.0

ASUS created on 29-Jan-23. Last modified 29-Jan-23

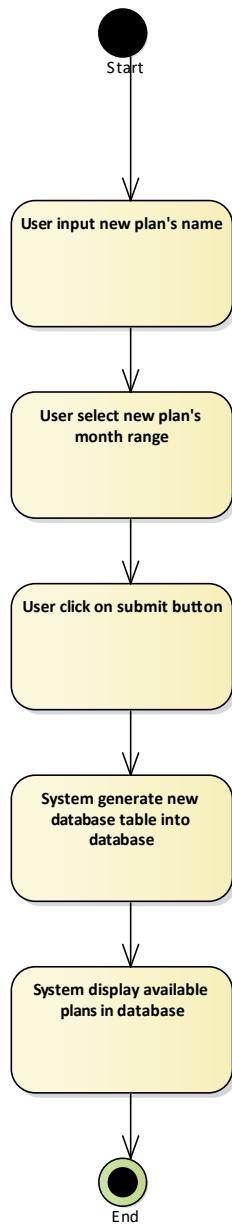


Figure 3: Create Plan_ActivityGraph

System display available plans in database

Activity owned by 'Create Plan_ActivityGraph', in package 'Use Case Model'

System display available plans in database
 Version 1.0 Phase 1.0 Proposed
 ASUS created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from System display available plans in database to End

INCOMING BEHAVIORAL RELATIONSHIPS

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from System generate new database table into database to System display available plans in database

System generate new database table into database

Activity owned by 'Create Plan_ActivityGraph', in package 'Use Case Model'

System generate new database table into database
Version 1.0 Phase 1.0 Proposed

ASUS created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from System generate new database table into database to System display available plans in database

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User click on submit button to System generate new database table into database

User click on submit button

Activity owned by 'Create Plan_ActivityGraph', in package 'Use Case Model'

User click on submit button
Version 1.0 Phase 1.0 Proposed
ASUS created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User click on submit button to System generate new database table into database

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User select new plan's month range to User click on submit button

User input new plan's name

Activity owned by 'Create Plan_ActivityGraph', in package 'Use Case Model'

User input new plan's name
Version 1.0 Phase 1.0 Proposed
ASUS created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User input new plan's name to User select new plan's month range

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from Start to User input new plan's name

User select new plan's month range

Activity owned by 'Create Plan_ActivityGraph', in package 'Use Case Model'

User select new plan's month range
Version 1.0 Phase 1.0 Proposed
ASUS created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

⇐ ControlFlow from User select new plan's month range to User click on submit button

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User input new plan's name to User select new plan's month range

End

ActivityFinal owned by 'Create Plan_ActivityGraph', in package 'Use Case Model'

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from System display available plans in database to End

Start

ActivityInitial owned by 'Create Plan_ActivityGraph', in package 'Use Case Model'

OUTGOING BEHAVIORAL RELATIONSHIPS

⇐ ControlFlow from Start to User input new plan's name

Create Plan_SequenceDiagram

Interaction owned by 'Create Plan', in package 'Use Case Model'

Create Plan_SequenceDiagram
Version 1.0 Phase 1.0 Proposed
ASUS created on 29-Jan-23. Last modified 29-Jan-23

ELEMENTS OWNED BY Create Plan_SequenceDiagram

ELEMENTS OWNED BY Create Plan_SequenceDiagram	
✉	HomePage : Sequence «boundary»
✉	MainController : Sequence «control»
✉	planDatabase : Sequence «entity»
✉	User : Sequence

1_Basic_Path diagram

Interaction diagram in package 'Use Case Model'

1_Basic_Path
Version 1.0

ASUS created on 29-Jan-23. Last modified 29-Jan-23

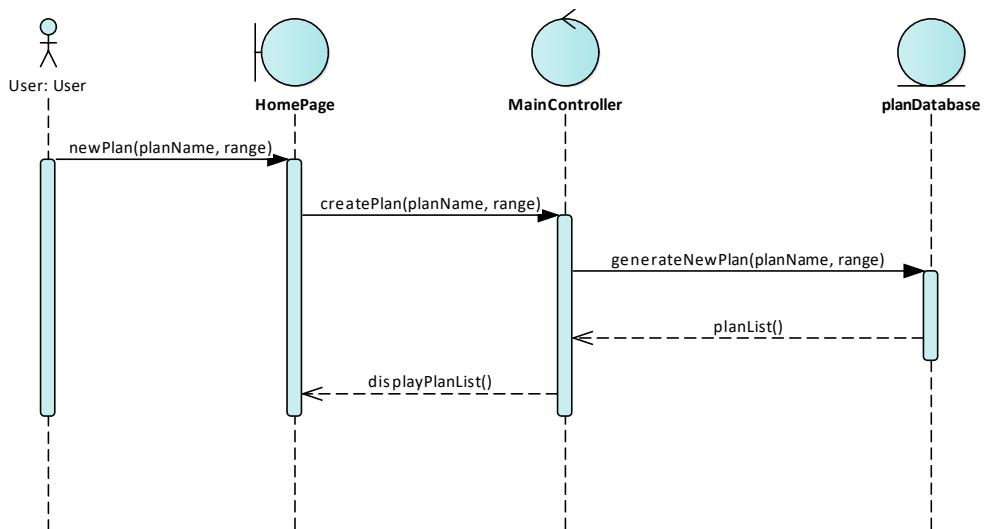


Figure 4: 1_Basic_Path

INTERACTION MESSAGES

- ✉ 1.0 'newPlan' from 'User: User' sent to 'HomePage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

- ✉ 1.1 'createPlan' from 'HomePage' sent to 'MainController'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.2 'generateNewPlan' from 'MainController' sent to 'planDatabase'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.3 'planList' from 'planDatabase' sent to 'MainController'.

Synchronous Call. Returns void.

[Return is True. Iteration is False. New group is False.]

1.4 'displayPlanList' from 'MainController' sent to 'HomePage'.

Synchronous Call. Returns void.

[Return is True. Iteration is False. New group is False.]

HomePage

Sequence «boundary» owned by 'Create Plan_SequenceDiagram', in package 'Use Case Model'

HomePage

Version 1.0 Phase 1.0 Proposed
ASUS created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: `createPlan`

Sequence from «boundary» HomePage to «control» MainController

INCOMING BEHAVIORAL RELATIONSHIPS

Name: `displayPlanList`

Sequence from «control» MainController to «boundary» HomePage

Name: `newPlan`

Sequence from User to «boundary» HomePage

MainController

Sequence «control» owned by 'Create Plan_SequenceDiagram', in package 'Use Case Model'

MainController

Version 1.0 Phase 1.0 Proposed
ASUS created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: `displayPlanList`

Sequence from «control» MainController to «boundary» HomePage

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: generateNewPlan
 ↳ Sequence from «control» MainController to «entity» planDatabase

INCOMING BEHAVIORAL RELATIONSHIPS

Name: planList
 ⇢ Sequence from «entity» planDatabase to «control» MainController

Name: createPlan
 ⇢ Sequence from «boundary» HomePage to «control» MainController

planDatabase

Sequence «entity» owned by 'Create Plan_SequenceDiagram', in package 'Use Case Model'

planDatabase
 Version 1.0 Phase 1.0 Proposed
 ASUS created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: planList
 ↳ Sequence from «entity» planDatabase to «control» MainController

INCOMING BEHAVIORAL RELATIONSHIPS

Name: generateNewPlan
 ⇢ Sequence from «control» MainController to «entity» planDatabase

User

Sequence owned by 'Create Plan_SequenceDiagram', in package 'Use Case Model'

User
 Version 1.0 Phase 1.0 Proposed
 ASUS created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: newPlan
 ↳ Sequence from User to «boundary» HomePage

Delete Plan

UseCase in package 'Use Case Model'

Delete Plan

ELEMENTS OWNED BY Delete Plan

 Delete Plan_ActivityGraph : Activity

 Delete Plan_SequenceDiagram : Interaction

SCENARIOS

 Basic Path. Basic Path

1. User clicked on "delete plan" button
2. System check whether selected plan is in the database
3. System delete selected plan in the database
4. System display all available plans in the database

CONSTRAINTS

 Pre-condition. User logged into the system

[Mandatory, Weight is 0.]

 Pre-condition. selected plan exist in database

[Mandatory, Weight is 1.]

 Post-condition. selected plan deleted from database

[Approved, Weight is 2.]

CONNECTORS

 UseCaseLink Source -> Destination

From: User : Actor, Public
To: Delete Plan : UseCase, Public

Delete Plan_ActivityGraph

Activity owned by 'Delete Plan', in package 'Use Case Model'

Delete Plan_ActivityGraph
Version 1.0 Phase 1.0 Proposed
ASUS created on 29-Jan-23. Last modified 29-Jan-23

ELEMENTS OWNED BY Delete Plan_ActivityGraph

 System check whether selected plan is in the database : Activity

 System delete selected plan in the database : Activity

ELEMENTS OWNED BY Delete Plan_ActivityGraph	
█	System display all available plans in the database : Activity
█	User clicked on "delete plan" button : Activity
█	End : ActivityFinal
█	Start : ActivityInitial

Delete Plan_ActivityGraph diagram

Activity diagram in package 'Use Case Model'

Delete Plan_ActivityGraph
Version 1.0

ASUS created on 29-Jan-23. Last modified 29-Jan-23

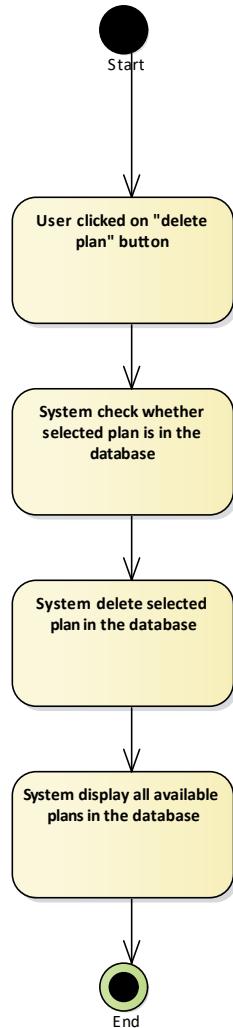


Figure 5: Delete Plan_ActivityGraph

System check whether selected plan is in the database

Activity owned by 'Delete Plan_ActivityGraph', in package 'Use Case Model'

System check whether selected plan is in the database
Version 1.0 Phase 1.0 Proposed
ASUS created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from System check whether selected plan is in the database to System delete selected plan in the database

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User clicked on "delete plan" button to System check whether selected plan is in the database

System delete selected plan in the database

Activity owned by 'Delete Plan_ActivityGraph', in package 'Use Case Model'

System delete selected plan in the database
Version 1.0 Phase 1.0 Proposed
ASUS created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from System delete selected plan in the database to System display all available plans in the database

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from System check whether selected plan is in the database to System delete selected plan in the database

System display all available plans in the database

Activity owned by 'Delete Plan_ActivityGraph', in package 'Use Case Model'

System display all available plans in the database
Version 1.0 Phase 1.0 Proposed
ASUS created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from System display all available plans in the database to End

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from System delete selected plan in the database to System display all available plans in the database

User clicked on "delete plan" button

Activity owned by 'Delete Plan_ActivityGraph', in package 'Use Case Model'

User clicked on "delete plan" button
 Version 1.0 Phase 1.0 Proposed
 ASUS created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from User clicked on "delete plan" button to System check whether selected plan is in the database

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from Start to User clicked on "delete plan" button

End

ActivityFinal owned by 'Delete Plan_ActivityGraph', in package 'Use Case Model'

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from System display all available plans in the database to End

Start

ActivityInitial owned by 'Delete Plan_ActivityGraph', in package 'Use Case Model'

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from Start to User clicked on "delete plan" button

Delete Plan_SequenceDiagram

Interaction owned by 'Delete Plan', in package 'Use Case Model'

Delete Plan_SequenceDiagram
 Version 1.0 Phase 1.0 Proposed
 ASUS created on 29-Jan-23. Last modified 29-Jan-23

ELEMENTS OWNED BY Delete Plan_SequenceDiagram

█ HistoryPage : Sequence «boundary»

█ mainController : Sequence «control»

ELEMENTS OWNED BY Delete Plan_SequenceDiagram	
█	planDatabase : Sequence «entity»
█	User : Sequence

1_Basic_Path diagram

Interaction diagram in package 'Use Case Model'

1_Basic_Path
Version 1.0

ASUS created on 29-Jan-23. Last modified 29-Jan-23

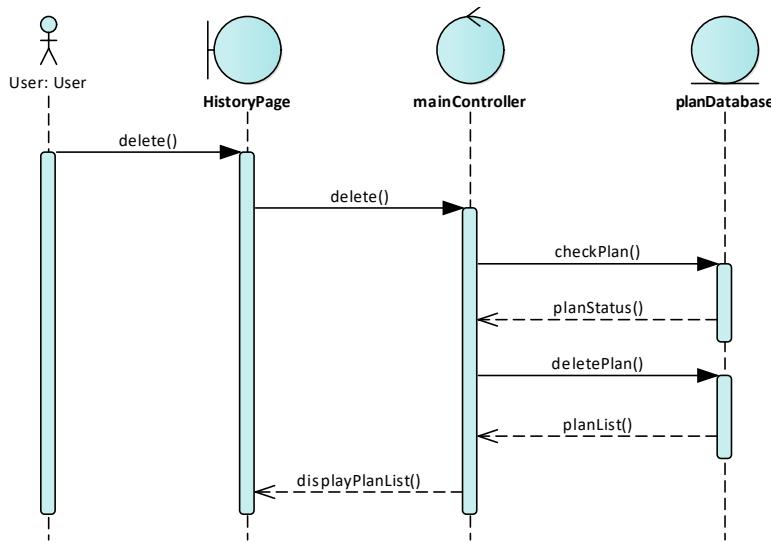


Figure 6: 1_Basic_Path

INTERACTION MESSAGES	
█	1.0 'delete() from 'User: User' sent to 'HistoryPage'.
	Synchronous Call. Returns void. [Return is False. Iteration is False. New group is False.]
█	1.1 'delete' from 'HistoryPage' sent to 'mainController'.
	Synchronous Call. Returns void. [Return is False. Iteration is False. New group is False.]
█	1.2 'checkPlan' from 'mainController' sent to 'planDatabase'.
	Synchronous Call. Returns void. [Return is False. Iteration is False. New group is False.]

1.3 'planStatus' from 'planDatabase' sent to 'mainController'.

Synchronous Call. Returns void.

[Return is True. Iteration is False. New group is False.]

1.4 'deletePlan' from 'mainController' sent to 'planDatabase'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.5 'planList' from 'planDatabase' sent to 'mainController'.

Synchronous Call. Returns void.

[Return is True. Iteration is False. New group is False.]

1.6 'displayPlanList' from 'mainController' sent to 'HistoryPage'.

Synchronous Call. Returns void.

[Return is True. Iteration is False. New group is False.]

HistoryPage

Sequence «boundary» owned by 'Delete Plan_SequenceDiagram', in package 'Use Case Model'

HistoryPage

Version 1.0 Phase 1.0 Proposed
ASUS created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: delete

Sequence from «boundary» HistoryPage to «control» mainController

INCOMING BEHAVIORAL RELATIONSHIPS

Name: displayPlanList

Sequence from «control» mainController to «boundary» HistoryPage

Name: delete()

Sequence from User to «boundary» HistoryPage

mainController

Sequence «control» owned by 'Delete Plan_SequenceDiagram', in package 'Use Case Model'

mainController

Version 1.0 Phase 1.0 Proposed
ASUS created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS	
Name: displayPlanList	Sequence from «control» mainController to «boundary» HistoryPage
Name: deletePlan	Sequence from «control» mainController to «entity» planDatabase
Name: checkPlan	Sequence from «control» mainController to «entity» planDatabase

INCOMING BEHAVIORAL RELATIONSHIPS	
Name: delete	Sequence from «boundary» HistoryPage to «control» mainController
Name: planStatus	Sequence from «entity» planDatabase to «control» mainController
Name: planList	Sequence from «entity» planDatabase to «control» mainController

planDatabase

Sequence «entity» owned by 'Delete Plan_SequenceDiagram', in package 'Use Case Model'

planDatabase
Version 1.0 Phase 1.0 Proposed
ASUS created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS	
Name: planStatus	Sequence from «entity» planDatabase to «control» mainController
Name: planList	Sequence from «entity» planDatabase to «control» mainController

INCOMING BEHAVIORAL RELATIONSHIPS	
Name: deletePlan	Sequence from «control» mainController to «entity» planDatabase
Name: checkPlan	Sequence from «control» mainController to «entity» planDatabase

User

Sequence owned by 'Delete Plan_SequenceDiagram', in package 'Use Case Model'

User

Version 1.0 Phase 1.0 Proposed
ASUS created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: delete()
 Sequence from User to «boundary» HistoryPage

Download Plan

UseCase in package 'Use Case Model'

Download Plan

Version 1.0 Phase 1.0 Proposed
ASUS created on 29-Jan-23. Last modified 29-Jan-23

ELEMENTS OWNED BY Download Plan

 Download Plan_SequenceDiagram : Interaction

SCENARIOS

 Basic Path. Basic Path

1. User clicked on the "download plan" button
2. System request data plan from the database
3. System get dataplan from the database
4. System generate the plan data in XLSX format

CONSTRAINTS

 Pre-condition. User has successfully logged into the system

[Mandatory, Weight is 0.]

 Pre-condition. plan existed in database

[Mandatory, Weight is 1.]

 Post-condition. User has successfully downloaded the file

[Mandatory, Weight is 2.]

CONNECTORS

 UseCaseLink Source -> Destination

From: User : Actor, Public
To: Download Plan : UseCase, Public

Download Plan_SequenceDiagram

Interaction owned by 'Download Plan', in package 'Use Case Model'

Download Plan_SequenceDiagram

Version 1.0 Phase 1.0 Proposed

USER created on 29-Jan-23. Last modified 29-Jan-23

ELEMENTS OWNED BY Download Plan_SequenceDiagram

HistoryPage : Sequence «boundary»

MainController : Sequence «control»

planDatabase : Sequence «entity»

User : Sequence

1_Basic_Path diagram

Interaction diagram in package 'Use Case Model'

1_Basic_Path
Version 1.0

USER created on 29-Jan-23. Last modified 29-Jan-23

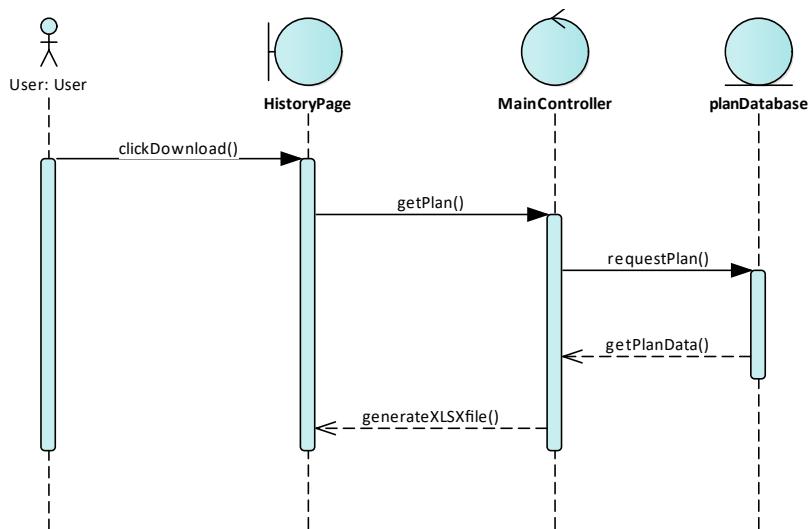


Figure 7: 1_Basic_Path

INTERACTION MESSAGES

1.0 'clickDownload' from 'User: User' sent to 'HistoryPage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.1 'getPlan' from 'HistoryPage' sent to 'MainController'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.2 'requestPlan' from 'MainController' sent to 'planDatabase'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.3 'getPlanData' from 'planDatabase' sent to 'MainController'.

Synchronous Call. Returns void.

[Return is True. Iteration is False. New group is False.]

1.4 'generateXLSXfile' from 'MainController' sent to 'HistoryPage'.

Synchronous Call. Returns void.

[Return is True. Iteration is False. New group is False.]

HistoryPage

Sequence «boundary» owned by 'Download Plan_SequenceDiagram', in package 'Use Case Model'

HistoryPage
Version 1.0 Phase 1.0 Proposed
USER created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: `getPlan`

Sequence from «boundary» HistoryPage to «control» MainController

INCOMING BEHAVIORAL RELATIONSHIPS

Name: `clickDownload`

Sequence from User to «boundary» HistoryPage

Name: `generateXLSXfile`

Sequence from «control» MainController to «boundary» HistoryPage

MainController

Sequence «control» owned by 'Download Plan_SequenceDiagram', in package 'Use Case Model'

MainController
Version 1.0 Phase 1.0 Proposed
USER created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: requestPlan
↳ Sequence from «control» MainController to «entity» planDatabase

Name: generateXLSXfile
↳ Sequence from «control» MainController to «boundary» HistoryPage

INCOMING BEHAVIORAL RELATIONSHIPS

Name: getPlan
⇒ Sequence from «boundary» HistoryPage to «control» MainController

Name: getPlanData
⇒ Sequence from «entity» planDatabase to «control» MainController

planDatabase

Sequence «entity» owned by 'Download Plan_SequenceDiagram', in package 'Use Case Model'

planDatabase
Version 1.0 Phase 1.0 Proposed
USER created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: getPlanData
↳ Sequence from «entity» planDatabase to «control» MainController

INCOMING BEHAVIORAL RELATIONSHIPS

Name: requestPlan
⇒ Sequence from «control» MainController to «entity» planDatabase

User

Sequence owned by 'Download Plan_SequenceDiagram', in package 'Use Case Model'

User
Version 1.0 Phase 1.0 Proposed
USER created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: clickDownload

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ Sequence from User to «boundary» HistoryPage

Input Demand

UseCase in package 'Use Case Model'

Input Demand
Version 1.0 Phase 1.0 Proposed
muhiq created on 13-Nov-22. Last modified 29-Jan-23

ELEMENTS OWNED BY Input Demand

█ Input Demand_ActivityGraph : Activity

█ Input Demand_SequenceDiagram : Interaction

SCENARIOS

█ Basic Path. Basic path

1. User clicks on the input demand button of a selected record
2. System displays the demand input form
3. User enters the monthly demands variables
4. User clicks on confirm
5. System processes the data inputs
6. System stores the demand input to the database

█ Exception. Month already exist

1. The system will display error

CONSTRAINTS

█ Post-condition. The monthly demand number has been inputted/modified

[Mandatory, Weight is 0.]

█ Pre-condition. The user is logged in to the system

[Mandatory, Weight is 1.]

CONNECTORS

↗ Extend «extend» Source -> Destination

From: View Optimized Plan : UseCase, Public
To: Input Demand : UseCase, Public

Input Demand_ActivityGraph

Activity owned by 'Input Demand', in package 'Use Case Model'

Input Demand_ActivityGraph
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

ELEMENTS OWNED BY Input Demand_ActivityGraph	
█	System displays the demand input form : Activity
█	System processes the data inputs : Activity
█	System stores the demand input to the database : Activity
█	User clicks on confirm : Activity
█	User clicks on the input demand button of a selected record : Activity
█	User enters the monthly demands variables : Activity
█	End : ActivityFinal
█	Start : ActivityInitial

Input Demand_ActivityGraph diagram

Activity diagram in package 'Use Case Model'

Input Demand_ActivityGraph
Version 1.0
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

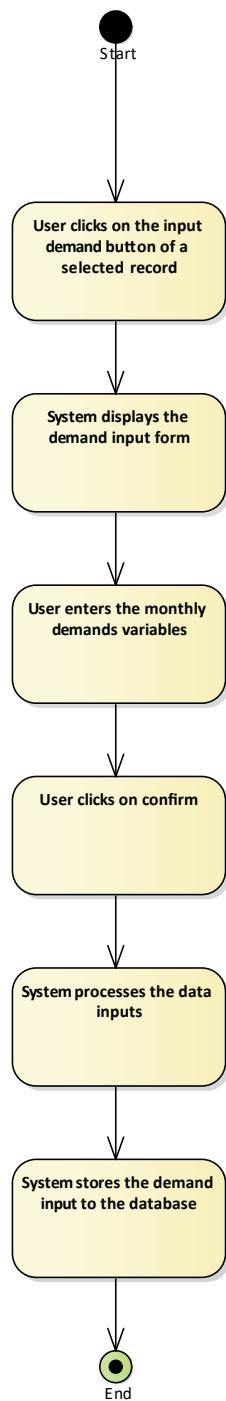


Figure 8: Input Demand_ActivityGraph

System displays the demand input form

Activity owned by 'Input Demand_ActivityGraph', in package 'Use Case Model'

System displays the demand input form
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from System displays the demand input form to User enters the monthly demands variables

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User clicks on the input demand button of a selected record to System displays the demand input form

System processes the data inputs

Activity owned by 'Input Demand_ActivityGraph', in package 'Use Case Model'

System processes the data inputs
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from System processes the data inputs to System stores the demand input to the database

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User clicks on confirm to System processes the data inputs

System stores the demand input to the database

Activity owned by 'Input Demand_ActivityGraph', in package 'Use Case Model'

System stores the demand input to the database
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from System stores the demand input to the database to End

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from System processes the data inputs to System stores the demand input to the database

User clicks on confirm

Activity owned by 'Input Demand_ActivityGraph', in package 'Use Case Model'

User clicks on confirm
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from User clicks on confirm to System processes the data inputs

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User enters the monthly demands variables to User clicks on confirm

User clicks on the input demand button of a selected record

Activity owned by 'Input Demand_ActivityGraph', in package 'Use Case Model'

User clicks on the input demand button of a selected record
 Version 1.0 Phase 1.0 Proposed
 KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from User clicks on the input demand button of a selected record to System displays the demand input form

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from Start to User clicks on the input demand button of a selected record

User enters the monthly demands variables

Activity owned by 'Input Demand_ActivityGraph', in package 'Use Case Model'

User enters the monthly demands variables
 Version 1.0 Phase 1.0 Proposed
 KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from User enters the monthly demands variables to User clicks on confirm

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from System displays the demand input form to User enters the monthly demands variables

End

ActivityFinal owned by 'Input Demand_ActivityGraph', in package 'Use Case Model'

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from System stores the demand input to the database to End

INCOMING BEHAVIORAL RELATIONSHIPS

Start

ActivityInitial owned by 'Input Demand_ActivityGraph', in package 'Use Case Model'

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from Start to User clicks on the input demand button of a selected record

Input Demand_SequenceDiagram

Interaction owned by 'Input Demand', in package 'Use Case Model'

Input Demand_SequenceDiagram

Version 1.0 Phase 1.0 Proposed

KingZexx created on 29-Jan-23. Last modified 29-Jan-23

ELEMENTS OWNED BY Input Demand_SequenceDiagram

█ DemandInputPage : Sequence «boundary»

█ HistoryPage : Sequence «boundary»

█ MainController : Sequence «control»

█ PlanDatabase : Sequence «entity»

█ PlanListPage : Sequence «boundary»

█ User : Sequence

1_Basic_Path diagram

Interaction diagram in package 'Use Case Model'

1_Basic_Path

Version 1.0

KingZexx created on 29-Jan-23. Last modified 30-Jan-23

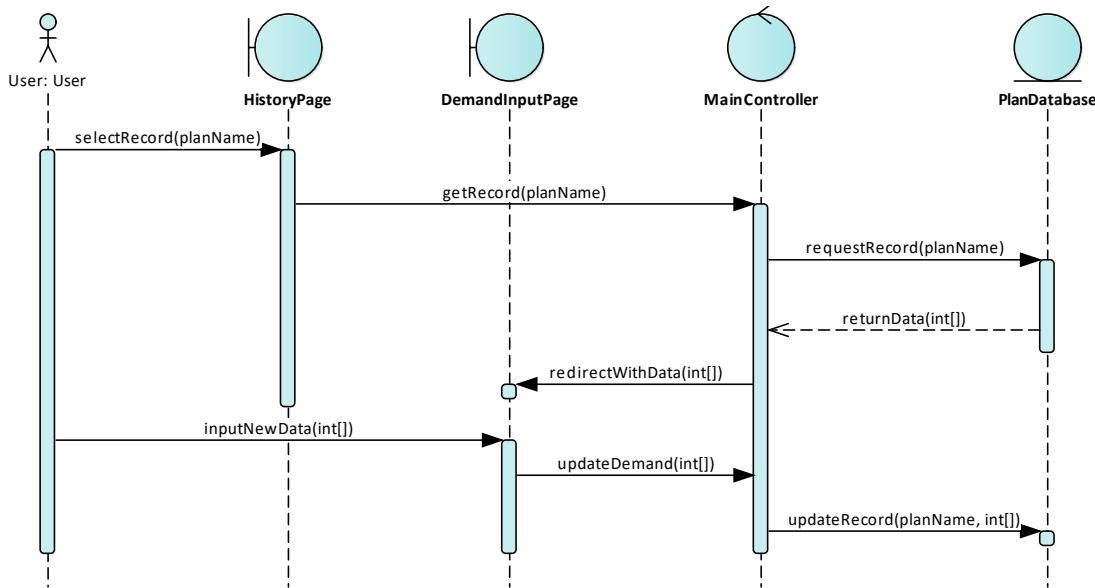


Figure 9: 1_Basic_Path

INTERACTION MESSAGES

- ✉ 1.0 'selectRecord(planName)' from 'User: User' sent to 'HistoryPage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

- ✉ 1.1 'getRecord' from 'HistoryPage' sent to 'MainController'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

- ✉ 1.2 'requestRecord' from 'MainController' sent to 'PlanDatabase'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

- ✉ 1.3 'returnData' from 'PlanDatabase' sent to 'MainController'.

Synchronous Call. Returns void.

[Return is True. Iteration is False. New group is False.]

- ✉ 1.4 'redirectWithData' from 'MainController' sent to 'DemandInputPage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

- ✉ 1.5 'inputNewData' from 'User: User' sent to 'DemandInputPage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.6 'updateDemand' from 'DemandInputPage' sent to 'MainController'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.7 'updateRecord' from 'MainController' sent to 'PlanDatabase'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

DemandInputPage

Sequence «boundary» owned by 'Input Demand_SequenceDiagram', in package 'Use Case Model'

DemandInputPage
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: updateDemand

Sequence from «boundary» DemandInputPage to «control» MainController

INCOMING BEHAVIORAL RELATIONSHIPS

Name: redirectWithData

Sequence from «control» MainController to «boundary» DemandInputPage

Name: inputNewData

Sequence from User to «boundary» DemandInputPage

HistoryPage

Sequence «boundary» owned by 'Input Demand_SequenceDiagram', in package 'Use Case Model'

HistoryPage
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: getRecord

Sequence from «boundary» HistoryPage to «control» MainController

INCOMING BEHAVIORAL RELATIONSHIPS

Name: selectRecord(planName)

Sequence from User to «boundary» HistoryPage

MainController

Sequence «control» owned by 'Input Demand_SequenceDiagram', in package 'Use Case Model'

MainController
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

- Name: requestRecord
 - ↳ Sequence from «control» MainController to «entity» PlanDatabase
- Name: redirectWithData
 - ↳ Sequence from «control» MainController to «boundary» DemandInputPage
- Name: updateRecord
 - ↳ Sequence from «control» MainController to «entity» PlanDatabase

INCOMING BEHAVIORAL RELATIONSHIPS

- Name: returnData
 - ⇒ Sequence from «entity» PlanDatabase to «control» MainController
- Name: getRecord
 - ⇒ Sequence from «boundary» HistoryPage to «control» MainController
- Name: updateDemand
 - ⇒ Sequence from «boundary» DemandInputPage to «control» MainController

PlanDatabase

Sequence «entity» owned by 'Input Demand_SequenceDiagram', in package 'Use Case Model'

PlanDatabase
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

- Name: returnData
 - ↳ Sequence from «entity» PlanDatabase to «control» MainController

INCOMING BEHAVIORAL RELATIONSHIPS

- Name: requestRecord
 - ⇒ Sequence from «control» MainController to «entity» PlanDatabase

INCOMING BEHAVIORAL RELATIONSHIPS

Name: updateRecord
 ↳ Sequence from «control» MainController to «entity» PlanDatabase

PlanListPage

Sequence «boundary» owned by 'Input Demand_SequenceDiagram', in package 'Use Case Model'

PlanListPage
 Version 1.0 Phase 1.0 Proposed
 KingZexx created on 29-Jan-23. Last modified 29-Jan-23

User

Sequence owned by 'Input Demand_SequenceDiagram', in package 'Use Case Model'

User
 Version 1.0 Phase 1.0 Proposed
 KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: selectRecord(planName)
 ↲ Sequence from User to «boundary» HistoryPage

Name: inputNewData
 ↲ Sequence from User to «boundary» DemandInputPage

Input Firing Cost

UseCase in package 'Use Case Model'

Input Firing Cost
 Version 1.0 Phase 1.0 Proposed
 muhiq created on 13-Nov-22. Last modified 29-Jan-23

ELEMENTS OWNED BY Input Firing Cost

▣ Input Firing Cost_ActivityGraph : Activity

▣ Input Firing Cost_SequenceDiagram : Interaction

SCENARIOS

▣ Basic Path. Basic Path

1. User clicks on the input firing cost button of a selected record
2. System displays the firing cost input form
3. User enters the firing cost variables
4. User clicks on confirm

SCENARIOS

- 5. System processes the data inputs
- 6. System stores the firing cost input to the database

Alternate. If No Exist Inventory Cost

1. <include> Calculate inventory holding cost

Alternate. If No Exist Hiring and Firing Cost

1. <include> Calculate firing and hiring cost

Alternate. Add month

1. The system displays "Adding Month from Scratch"

CONSTRAINTS

Pre-condition. The user is logged in to the system

[Approved, Weight is 0.]

Post-condition. The firing cost has been inputted/modified

[Approved, Weight is 1.]

CONNECTORS

Extend «extend» Source -> Destination
 From: View Optimized Plan : UseCase, Public
 To: Input Firing Cost : UseCase, Public

Input Firing Cost_ActivityGraph

Activity owned by 'Input Firing Cost', in package 'Use Case Model'

Input Firing Cost_ActivityGraph
 Version 1.0 Phase 1.0 Proposed
 KingZexx created on 29-Jan-23. Last modified 29-Jan-23

ELEMENTS OWNED BY Input Firing Cost_ActivityGraph

System displays the firing cost input form : Activity

System processes the data inputs : Activity

System stores the firing cost input to the database : Activity

ELEMENTS OWNED BY Input Firing Cost_ActivityGraph	
█	User clicks on confirm : Activity
█	User clicks on the input firing cost button of a selected record : Activity
█	User enters the firing cost variables : Activity
█	End : ActivityFinal
█	Start : ActivityInitial

Input Firing Cost_ActivityGraph diagram

Activity diagram in package 'Use Case Model'

Input Firing Cost_ActivityGraph
Version 1.0
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

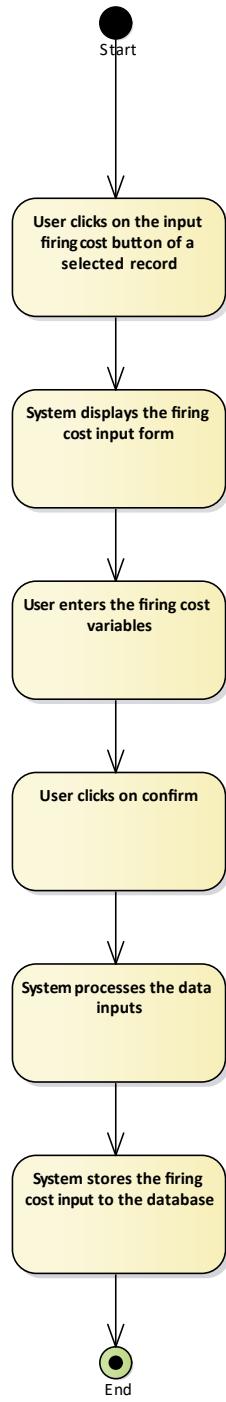


Figure 10: Input Firing Cost_ActivityGraph

System displays the firing cost input form

Activity owned by 'Input Firing Cost_ActivityGraph', in package 'Use Case Model'

System displays the firing cost input form
 Version 1.0 Phase 1.0 Proposed
 KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from System displays the firing cost input form to User enters the firing cost variables

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User clicks on the input firing cost button of a selected record to System displays the firing cost input form

System processes the data inputs

Activity owned by 'Input Firing Cost_ActivityGraph', in package 'Use Case Model'

System processes the data inputs
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from System processes the data inputs to System stores the firing cost input to the database

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User clicks on confirm to System processes the data inputs

System stores the firing cost input to the database

Activity owned by 'Input Firing Cost_ActivityGraph', in package 'Use Case Model'

System stores the firing cost input to the database
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from System stores the firing cost input to the database to End

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from System processes the data inputs to System stores the firing cost input to the database

User clicks on confirm

Activity owned by 'Input Firing Cost_ActivityGraph', in package 'Use Case Model'

User clicks on confirm
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from User clicks on confirm to System processes the data inputs

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User enters the firing cost variables to User clicks on confirm

User clicks on the input firing cost button of a selected record

Activity owned by 'Input Firing Cost_ActivityGraph', in package 'Use Case Model'

User clicks on the input firing cost button of a selected record
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from User clicks on the input firing cost button of a selected record to System displays the firing cost input form

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from Start to User clicks on the input firing cost button of a selected record

User enters the firing cost variables

Activity owned by 'Input Firing Cost_ActivityGraph', in package 'Use Case Model'

User enters the firing cost variables
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from User enters the firing cost variables to User clicks on confirm

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from System displays the firing cost input form to User enters the firing cost variables

End

ActivityFinal owned by 'Input Firing Cost_ActivityGraph', in package 'Use Case Model'

INCOMING BEHAVIORAL RELATIONSHIPS

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from System stores the firing cost input to the database to End

Start

ActivityInitial owned by 'Input Firing Cost_ActivityGraph', in package 'Use Case Model'

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from Start to User clicks on the input firing cost button of a selected record

Input Firing Cost_SequenceDiagram

Interaction owned by 'Input Firing Cost', in package 'Use Case Model'

Input Firing Cost_SequenceDiagram

Version 1.0 Phase 1.0 Proposed

KingZexx created on 29-Jan-23. Last modified 29-Jan-23

ELEMENTS OWNED BY Input Firing Cost_SequenceDiagram

- █ FiringCostInputPage : Sequence «boundary»
- █ HistoryPage : Sequence «boundary»
- █ MainController : Sequence «control»
- █ PlanDatabase : Sequence «entity»
- █ User : Sequence

1_Basic_Path diagram

Interaction diagram in package 'Use Case Model'

1_Basic_Path

Version 1.0

KingZexx created on 29-Jan-23. Last modified 29-Jan-23

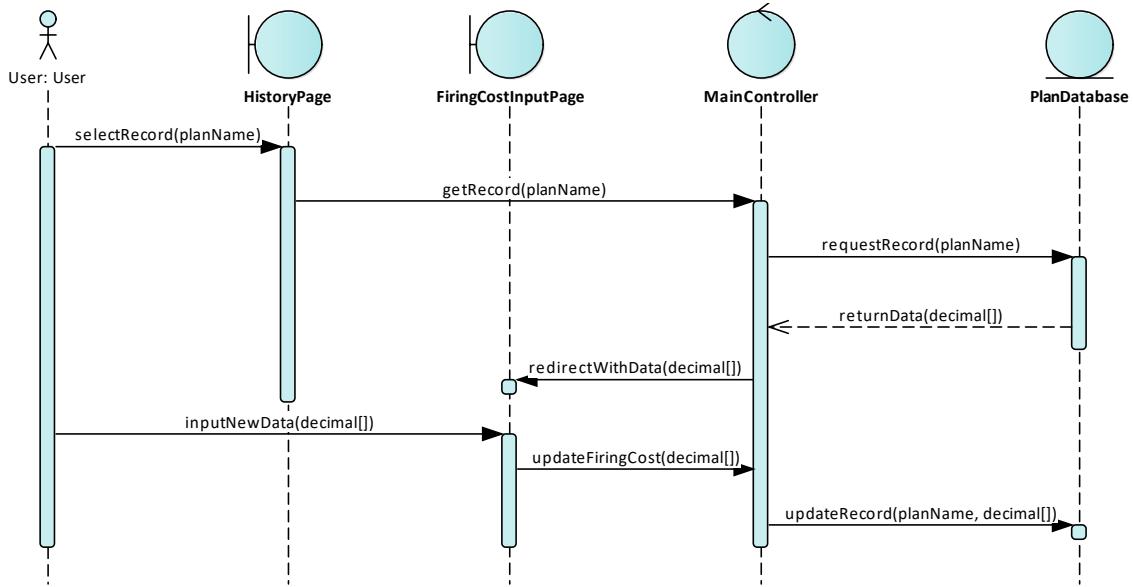


Figure 11: 1_Basic_Path

INTERACTION MESSAGES

- ✉ 1.0 'selectRecord(planName)' from 'User: User' sent to 'HistoryPage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

- ✉ 1.1 'getRecord' from 'HistoryPage' sent to 'MainController'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

- ✉ 1.2 'requestRecord' from 'MainController' sent to 'PlanDatabase'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

- ✉ 1.3 'returnData' from 'PlanDatabase' sent to 'MainController'.

Synchronous Call. Returns void.

[Return is True. Iteration is False. New group is False.]

- ✉ 1.4 'redirectWithData' from 'MainController' sent to 'FiringCostInputPage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

- ✉ 1.5 'inputNewData' from 'User: User' sent to 'FiringCostInputPage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.6 'updateFiringCost' from 'FiringCostInputPage' sent to 'MainController'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.7 'updateRecord' from 'MainController' sent to 'PlanDatabase'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

FiringCostInputPage

Sequence «boundary» owned by 'Input Firing Cost_SequenceDiagram', in package 'Use Case Model'

FiringCostInputPage
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: updateFiringCost

↳ Sequence from «boundary» FiringCostInputPage to «control» MainController

INCOMING BEHAVIORAL RELATIONSHIPS

Name: redirectWithData

↗ Sequence from «control» MainController to «boundary» FiringCostInputPage

Name: inputNewData

↗ Sequence from User to «boundary» FiringCostInputPage

HistoryPage

Sequence «boundary» owned by 'Input Firing Cost_SequenceDiagram', in package 'Use Case Model'

HistoryPage
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: getRecord

↳ Sequence from «boundary» HistoryPage to «control» MainController

INCOMING BEHAVIORAL RELATIONSHIPS

Name: selectRecord(planName)

↗ Sequence from User to «boundary» HistoryPage

MainController

Sequence «control» owned by 'Input Firing Cost_SequenceDiagram', in package 'Use Case Model'

MainController
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: requestRecord
 Sequence from «control» MainController to «entity» PlanDatabase

Name: redirectWithData
 Sequence from «control» MainController to «boundary» FiringCostInputPage

Name: updateRecord
 Sequence from «control» MainController to «entity» PlanDatabase

INCOMING BEHAVIORAL RELATIONSHIPS

Name: getRecord
 Sequence from «boundary» HistoryPage to «control» MainController

Name: returnData
 Sequence from «entity» PlanDatabase to «control» MainController

Name: updateFiringCost
 Sequence from «boundary» FiringCostInputPage to «control» MainController

PlanDatabase

Sequence «entity» owned by 'Input Firing Cost_SequenceDiagram', in package 'Use Case Model'

PlanDatabase
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: returnData
 Sequence from «entity» PlanDatabase to «control» MainController

INCOMING BEHAVIORAL RELATIONSHIPS

Name: requestRecord
 Sequence from «control» MainController to «entity» PlanDatabase

INCOMING BEHAVIORAL RELATIONSHIPS

Name: updateRecord

➡ Sequence from «control» MainController to «entity» PlanDatabase

User*Sequence owned by 'Input Firing Cost_SequenceDiagram', in package 'Use Case Model'*

User

Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23**OUTGOING BEHAVIORAL RELATIONSHIPS**

Name: selectRecord(planName)

⬅ Sequence from User to «boundary» HistoryPage

Name: inputNewData

⬅ Sequence from User to «boundary» FiringCostInputPage

Input Hiring Cost*UseCase in package 'Use Case Model'*

Input Hiring Cost

Version 1.0 Phase 1.0 Proposed
muhiq created on 13-Nov-22. Last modified 29-Jan-23**ELEMENTS OWNED BY Input Hiring Cost**

█ Input Hiring Cost_ActivityGraph : Activity

█ Input Hiring Cost_SequenceDiagram : Interaction

SCENARIOS

█ Basic Path. Basic Path

1. User clicks on the input hiring cost button of a selected record
2. System display the hiring cost input form
3. User enters the hiring cost variables
4. User clicks on confirm
5. System processes the data inputs
6. System stores the hiring cost input to the database

█ Alternate. Si- is not Equal or More Than 0

1. The system display "The number of Temporary Workers Fired is not accepted"

SCENARIOS

Alternate. Si+ is not Equal or More Than 0

1. The system display "The number of Temporary Workers Hired is not accepted"

CONSTRAINTS

Pre-condition. The user is logged in to the system

[Mandatory, Weight is 0.]

Post-condition. The hiring cost number has been inputted/modified

[Mandatory, Weight is 1.]

CONNECTORS

Extend «extend» Source -> Destination

From: View Optimized Plan : UseCase, Public
To: Input Hiring Cost : UseCase, Public

Input Hiring Cost_ActivityGraph

Activity owned by 'Input Hiring Cost', in package 'Use Case Model'

Input Hiring Cost_ActivityGraph
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

ELEMENTS OWNED BY Input Hiring Cost_ActivityGraph

System display the hiring cost input form : Activity

System processes the data inputs : Activity

System stores the hiring cost input to the database : Activity

User clicks on confirm : Activity

User clicks on the input hiring cost button of a selected record : Activity

User enters the hiring cost variables : Activity

End : ActivityFinal

Start : ActivityInitial

Input Hiring Cost_ActivityGraph diagram

Activity diagram in package 'Use Case Model'

Input Hiring Cost_ActivityGraph

Version 1.0

KingZexx created on 29-Jan-23. Last modified 29-Jan-23

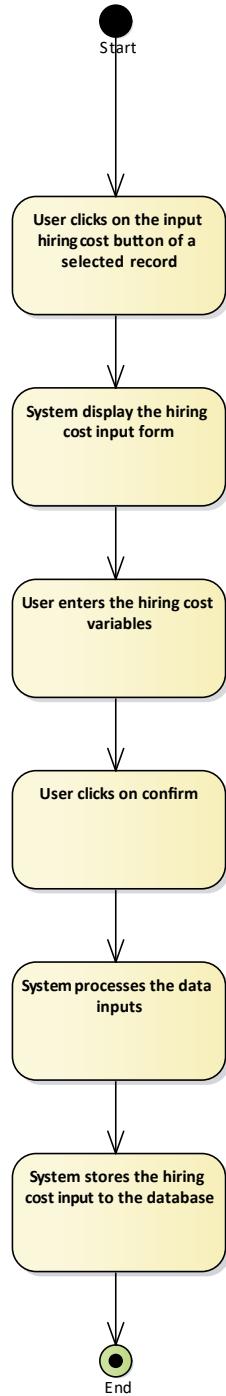


Figure 12: Input Hiring Cost_ActivityGraph

System display the hiring cost input form

Activity owned by 'Input Hiring Cost_ActivityGraph', in package 'Use Case Model'

System display the hiring cost input form
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from System display the hiring cost input form to User enters the hiring cost variables

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User clicks on the input hiring cost button of a selected record to System display the hiring cost input form

System processes the data inputs

Activity owned by 'Input Hiring Cost_ActivityGraph', in package 'Use Case Model'

System processes the data inputs
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from System processes the data inputs to System stores the hiring cost input to the database

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User clicks on confirm to System processes the data inputs

System stores the hiring cost input to the database

Activity owned by 'Input Hiring Cost_ActivityGraph', in package 'Use Case Model'

System stores the hiring cost input to the database
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from System stores the hiring cost input to the database to End

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from System processes the data inputs to System stores the hiring cost input to the database

User clicks on confirm

Activity owned by 'Input Hiring Cost_ActivityGraph', in package 'Use Case Model'

User clicks on confirm
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from User clicks on confirm to System processes the data inputs

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User enters the hiring cost variables to User clicks on confirm

User clicks on the input hiring cost button of a selected record

Activity owned by 'Input Hiring Cost_ActivityGraph', in package 'Use Case Model'

User clicks on the input hiring cost button of a selected record
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from User clicks on the input hiring cost button of a selected record to System display the hiring cost input form

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from Start to User clicks on the input hiring cost button of a selected record

User enters the hiring cost variables

Activity owned by 'Input Hiring Cost_ActivityGraph', in package 'Use Case Model'

User enters the hiring cost variables
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from User enters the hiring cost variables to User clicks on confirm

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from System display the hiring cost input form to User enters the hiring cost variables

End

ActivityFinal owned by 'Input Hiring Cost_ActivityGraph', in package 'Use Case Model'

INCOMING BEHAVIORAL RELATIONSHIPS

- ⇒ ControlFlow from System stores the hiring cost input to the database to End

Start

ActivityInitial owned by 'Input Hiring Cost_ActivityGraph', in package 'Use Case Model'

OUTGOING BEHAVIORAL RELATIONSHIPS

- ⇐ ControlFlow from Start to User clicks on the input hiring cost button of a selected record

Input Hiring Cost_SequenceDiagram

Interaction owned by 'Input Hiring Cost', in package 'Use Case Model'

Input Hiring Cost_SequenceDiagram

Version 1.0 Phase 1.0 Proposed

KingZexx created on 29-Jan-23. Last modified 29-Jan-23

ELEMENTS OWNED BY Input Hiring Cost_SequenceDiagram

- █ HiringCostInputPage : Sequence «boundary»
- █ HistoryPage : Sequence «boundary»
- █ MainController : Sequence «control»
- █ PlanDatabase : Sequence «entity»
- █ User : Sequence

1_Basic_Path diagram

Interaction diagram in package 'Use Case Model'

1_Basic_Path

Version 1.0

KingZexx created on 29-Jan-23. Last modified 29-Jan-23

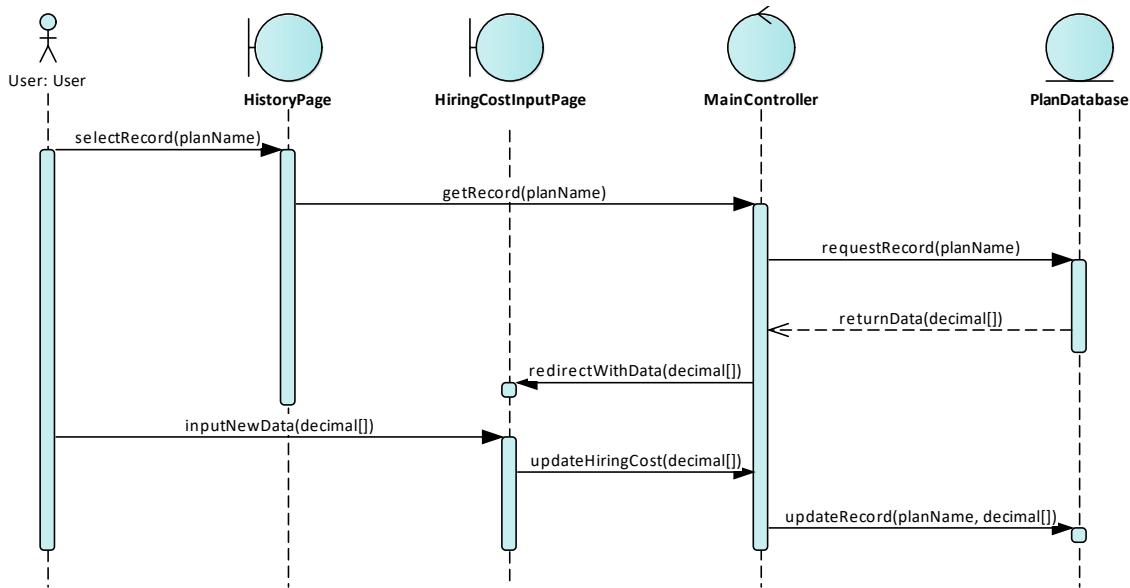


Figure 13: 1_Basic_Path

INTERACTION MESSAGES

- ✉ 1.0 'selectRecord(planName)' from 'User: User' sent to 'HistoryPage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

- ✉ 1.1 'getRecord' from 'HistoryPage' sent to 'MainController'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

- ✉ 1.2 'requestRecord' from 'MainController' sent to 'PlanDatabase'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

- ✉ 1.3 'returnData' from 'PlanDatabase' sent to 'MainController'.

Synchronous Call. Returns void.

[Return is True. Iteration is False. New group is False.]

- ✉ 1.4 'redirectWithData' from 'MainController' sent to 'HiringCostInputPage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

- ✉ 1.5 'inputNewData' from 'User: User' sent to 'HiringCostInputPage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.6 'updateHiringCost' from 'HiringCostInputPage' sent to 'MainController'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.7 'updateRecord' from 'MainController' sent to 'PlanDatabase'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

HiringCostInputPage

Sequence «boundary» owned by 'Input Hiring Cost_SequenceDiagram', in package 'Use Case Model'

HiringCostInputPage
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: updateHiringCost

Sequence from «boundary» HiringCostInputPage to «control» MainController

INCOMING BEHAVIORAL RELATIONSHIPS

Name: redirectWithData

Sequence from «control» MainController to «boundary» HiringCostInputPage

Name: inputNewData

Sequence from User to «boundary» HiringCostInputPage

HistoryPage

Sequence «boundary» owned by 'Input Hiring Cost_SequenceDiagram', in package 'Use Case Model'

HistoryPage
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: getRecord

Sequence from «boundary» HistoryPage to «control» MainController

INCOMING BEHAVIORAL RELATIONSHIPS

Name: selectRecord(planName)

Sequence from User to «boundary» HistoryPage

MainController

Sequence «control» owned by 'Input Hiring Cost_SequenceDiagram', in package 'Use Case Model'

MainController
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: requestRecord
 Sequence from «control» MainController to «entity» PlanDatabase

Name: updateRecord
 Sequence from «control» MainController to «entity» PlanDatabase

Name: redirectWithData
 Sequence from «control» MainController to «boundary» HiringCostInputPage

INCOMING BEHAVIORAL RELATIONSHIPS

Name: returnData
 Sequence from «entity» PlanDatabase to «control» MainController

Name: getRecord
 Sequence from «boundary» HistoryPage to «control» MainController

Name: updateHiringCost
 Sequence from «boundary» HiringCostInputPage to «control» MainController

PlanDatabase

Sequence «entity» owned by 'Input Hiring Cost_SequenceDiagram', in package 'Use Case Model'

PlanDatabase
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: returnData
 Sequence from «entity» PlanDatabase to «control» MainController

INCOMING BEHAVIORAL RELATIONSHIPS

Name: requestRecord
 Sequence from «control» MainController to «entity» PlanDatabase

INCOMING BEHAVIORAL RELATIONSHIPS

Name: updateRecord

➡ Sequence from «control» MainController to «entity» PlanDatabase

User*Sequence owned by 'Input Hiring Cost_SequenceDiagram', in package 'Use Case Model'*

User

Version 1.0 Phase 1.0 Proposed

KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: selectRecord(planName)

⬅ Sequence from User to «boundary» HistoryPage

Name: inputNewData

⬅ Sequence from User to «boundary» HiringCostInputPage

Input Holding Cost per Unit*UseCase in package 'Use Case Model'*

Input Holding Cost per Unit

Version 1.0 Phase 1.0 Proposed

muhiq created on 13-Nov-22. Last modified 29-Jan-23

ELEMENTS OWNED BY Input Holding Cost per Unit

█ Input Holding Cost per Unit_ActivityGraph : Activity

█ Input Holding Cost per Unit_SequenceDiagram : Interaction

SCENARIOS

█ Basic Path. Basic Path

1. User clicks on the input holding cost per unit button of a selected record
2. System displays the holding cost per unit input form
3. User enters the holding cost per unit variables
4. User clicks on confirm
5. System processes the data inputs
6. System stores the holding cost per unit input to the database

CONSTRAINTS

█ Pre-condition. The user is logged in to the system

[Mandatory, Weight is 0.]

CONSTRAINTS

- ⌚ Post-condition. The holding cost per unit number has been inputted/modified

[Mandatory, Weight is 1.]

CONNECTORS

-  **Extend** «extend» Source -> Destination
 From: View Optimized Plan : UseCase, Public
 To: Input Holding Cost per Unit : UseCase, Public

Input Holding Cost per Unit_ActivityGraph

Activity owned by 'Input Holding Cost per Unit', in package 'Use Case Model'

Input Holding Cost per Unit_ActivityGraph
 Version 1.0 Phase 1.0 Proposed
 KingZexx created on 29-Jan-23. Last modified 29-Jan-23

ELEMENTS OWNED BY Input Holding Cost per Unit_ActivityGraph

- ▣ System displays the holding cost per unit input form : Activity
- ▣ System processes the data inputs : Activity
- ▣ System stores the holding cost per unit input to the database : Activity
- ▣ User clicks on confirm : Activity
- ▣ User clicks on the input holding cost per unit button of a selected record : Activity
- ▣ User enters the holding cost per unit variables : Activity
- ▣ End : ActivityFinal
- ▣ Start : ActivityInitial

Input Holding Cost per Unit_ActivityGraph diagram

Activity diagram in package 'Use Case Model'

Input Holding Cost per Unit_ActivityGraph
 Version 1.0
 KingZexx created on 29-Jan-23. Last modified 29-Jan-23

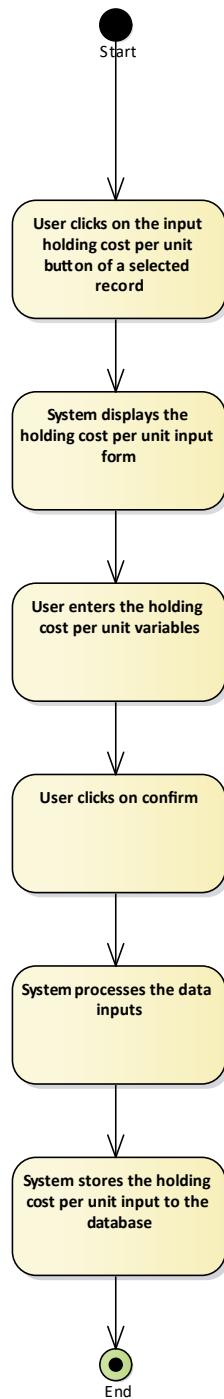


Figure 14: Input Holding Cost per Unit_ActivityGraph

System displays the holding cost per unit input form

Activity owned by 'Input Holding Cost per Unit_ActivityGraph', in package 'Use Case Model'

System displays the holding cost per unit input form
 Version 1.0 Phase 1.0 Proposed
 KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from System displays the holding cost per unit input form to User enters the holding cost per unit variables

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User clicks on the input holding cost per unit button of a selected record to System displays the holding cost per unit input form

System processes the data inputs

Activity owned by 'Input Holding Cost per Unit_ActivityGraph', in package 'Use Case Model'

System processes the data inputs
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from System processes the data inputs to System stores the holding cost per unit input to the database

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User clicks on confirm to System processes the data inputs

System stores the holding cost per unit input to the database

Activity owned by 'Input Holding Cost per Unit_ActivityGraph', in package 'Use Case Model'

System stores the holding cost per unit input to the database
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from System stores the holding cost per unit input to the database to End

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from System processes the data inputs to System stores the holding cost per unit input to the database

User clicks on confirm

Activity owned by 'Input Holding Cost per Unit_ActivityGraph', in package 'Use Case Model'

User clicks on confirm
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from User clicks on confirm to System processes the data inputs

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User enters the holding cost per unit variables to User clicks on confirm

User clicks on the input holding cost per unit button of a selected record

Activity owned by 'Input Holding Cost per Unit_ActivityGraph', in package 'Use Case Model'

User clicks on the input holding cost per unit button of a selected record
 Version 1.0 Phase 1.0 Proposed
 KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from User clicks on the input holding cost per unit button of a selected record to System displays the holding cost per unit input form

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from Start to User clicks on the input holding cost per unit button of a selected record

User enters the holding cost per unit variables

Activity owned by 'Input Holding Cost per Unit_ActivityGraph', in package 'Use Case Model'

User enters the holding cost per unit variables
 Version 1.0 Phase 1.0 Proposed
 KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from User enters the holding cost per unit variables to User clicks on confirm

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from System displays the holding cost per unit input form to User enters the holding cost per unit variables

End

ActivityFinal owned by 'Input Holding Cost per Unit_ActivityGraph', in package 'Use Case Model'

INCOMING BEHAVIORAL RELATIONSHIPS

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from System stores the holding cost per unit input to the database to End

Start

ActivityInitial owned by 'Input Holding Cost per Unit_ActivityGraph', in package 'Use Case Model'

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from Start to User clicks on the input holding cost per unit button of a selected record

Input Holding Cost per Unit_SequenceDiagram

Interaction owned by 'Input Holding Cost per Unit', in package 'Use Case Model'

Input Holding Cost per Unit_SequenceDiagram

Version 1.0 Phase 1.0 Proposed

KingZexx created on 29-Jan-23. Last modified 29-Jan-23

ELEMENTS OWNED BY Input Holding Cost per Unit_SequenceDiagram

█ HistoryPage : Sequence «boundary»

█ HoldingCostInputPage : Sequence «boundary»

█ MainController : Sequence «control»

█ PlanDatabase : Sequence «entity»

█ User : Sequence

1_Basic_Path diagram

Interaction diagram in package 'Use Case Model'

1_Basic_Path

Version 1.0

KingZexx created on 29-Jan-23. Last modified 29-Jan-23

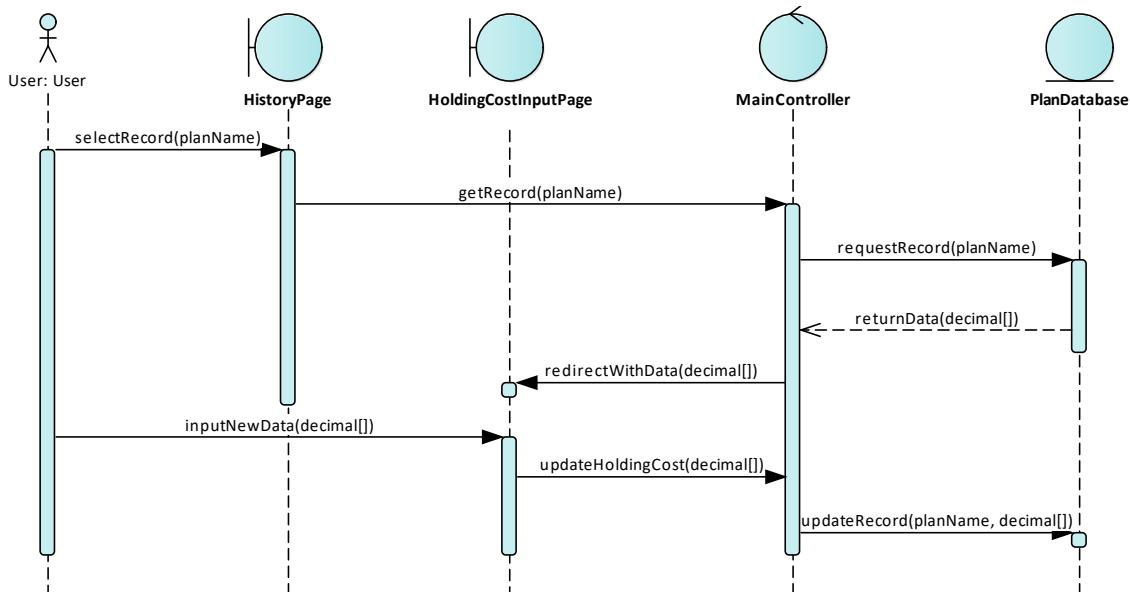


Figure 15: 1_Basic_Path

INTERACTION MESSAGES

- ✉ 1.0 'selectRecord(planName)' from 'User: User' sent to 'HistoryPage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

- ✉ 1.1 'getRecord' from 'HistoryPage' sent to 'MainController'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

- ✉ 1.2 'requestRecord' from 'MainController' sent to 'PlanDatabase'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

- ✉ 1.3 'returnData' from 'PlanDatabase' sent to 'MainController'.

Synchronous Call. Returns void.

[Return is True. Iteration is False. New group is False.]

- ✉ 1.4 'redirectWithData' from 'MainController' sent to 'HoldingCostInputPage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

- ✉ 1.5 'inputNewData' from 'User: User' sent to 'HoldingCostInputPage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.6 'updateHoldingCost' from 'HoldingCostInputPage' sent to 'MainController'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.7 'updateRecord' from 'MainController' sent to 'PlanDatabase'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

HistoryPage

Sequence «boundary» owned by 'Input Holding Cost per Unit_SequenceDiagram', in package 'Use Case Model'

HistoryPage

Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: getRecord

Sequence from «boundary» HistoryPage to «control» MainController

INCOMING BEHAVIORAL RELATIONSHIPS

Name: selectRecord(planName)

Sequence from User to «boundary» HistoryPage

HoldingCostInputPage

Sequence «boundary» owned by 'Input Holding Cost per Unit_SequenceDiagram', in package 'Use Case Model'

HoldingCostInputPage

Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: updateHoldingCost

Sequence from «boundary» HoldingCostInputPage to «control» MainController

INCOMING BEHAVIORAL RELATIONSHIPS

Name: redirectWithData

Sequence from «control» MainController to «boundary» HoldingCostInputPage

Name: inputNewData

Sequence from User to «boundary» HoldingCostInputPage

MainController

Sequence «control» owned by 'Input Holding Cost per Unit_SequenceDiagram', in package 'Use Case Model'

MainController
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: requestRecord
 Sequence from «control» MainController to «entity» PlanDatabase

Name: updateRecord
 Sequence from «control» MainController to «entity» PlanDatabase

Name: redirectWithData
 Sequence from «control» MainController to «boundary» HoldingCostInputPage

INCOMING BEHAVIORAL RELATIONSHIPS

Name: returnData
 Sequence from «entity» PlanDatabase to «control» MainController

Name: updateHoldingCost
 Sequence from «boundary» HoldingCostInputPage to «control» MainController

Name: getRecord
 Sequence from «boundary» HistoryPage to «control» MainController

PlanDatabase

Sequence «entity» owned by 'Input Holding Cost per Unit_SequenceDiagram', in package 'Use Case Model'

PlanDatabase
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: returnData
 Sequence from «entity» PlanDatabase to «control» MainController

INCOMING BEHAVIORAL RELATIONSHIPS

Name: requestRecord
 Sequence from «control» MainController to «entity» PlanDatabase

INCOMING BEHAVIORAL RELATIONSHIPS

Name: updateRecord

➡ Sequence from «control» MainController to «entity» PlanDatabase

User*Sequence owned by 'Input Holding Cost per Unit_SequenceDiagram', in package 'Use Case Model'*

User

Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23**OUTGOING BEHAVIORAL RELATIONSHIPS**

Name: selectRecord(planName)

⬅ Sequence from User to «boundary» HistoryPage

Name: inputNewData

⬅ Sequence from User to «boundary» HoldingCostInputPage

Input Number of Permanent Worker*UseCase in package 'Use Case Model'*

Input Number of Permanent Worker

Version 1.0 Phase 1.0 Proposed
muhiq created on 13-Nov-22. Last modified 29-Jan-23**ELEMENTS OWNED BY Input Number of Permanent Worker**

█ Input Number of Permanent Worker_ActivityGraph : Activity

█ Input Number of Permanent Worker_SequenceDiagram : Interaction

SCENARIOS

█ Basic Path. Basic Path

1. User clicks on the input number of permanent worker button of a selected record
2. System displays the number of permanent worker input form
3. User enters the number of permanent worker variables
4. User clicks on confirm
5. System processes the data inputs
6. System stores the number of permanent worker input to the database

█ Alternate. Does it exist?

1. <include> Calc Rem. Demand Monthly

SCENARIOS	
Alternate.	Exist?
1. <include>	Calc. Number of Temporary Workers monthly
Alternate.	Is (Ii) more or equal than 0
1. Enter Units of ending inventory for (Ii)	
Alternate.	Is (Ii-1) more or equal than zero
1. Enter Units of ending inventory for last month (Ii-1)	

CONSTRAINTS	
Pre-condition.	The user is logged in to the system [Mandatory, Weight is 0.]
Post-condition.	The number of permanent worker data has been inputted/modified [Mandatory, Weight is 1.]

CONNECTORS	
Extend	«extend» Source -> Destination From: View Optimized Plan : UseCase, Public To: Input Number of Permanent Worker : UseCase, Public

Input Number of Permanent Worker_ActivityGraph

Activity owned by 'Input Number of Permanent Worker', in package 'Use Case Model'

Input Number of Permanent Worker_ActivityGraph
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

ELEMENTS OWNED BY Input Number of Permanent Worker_ActivityGraph	
	System displays the number of permanent worker input form : Activity
	System processes the data inputs : Activity
	System stores the number of permanent worker input to the database : Activity
	User clicks on confirm : Activity
	User clicks on the input number of permanent worker button of a selected record : Activity

ELEMENTS OWNED BY Input Number of Permanent Worker_ActivityGraph
>User enters the number of permanent worker variables : Activity
End : ActivityFinal
Start : ActivityInitial

Input Number of Permanent Worker_ActivityGraph diagram

Activity diagram in package 'Use Case Model'

Input Number of Permanent Worker_ActivityGraph
Version 1.0
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

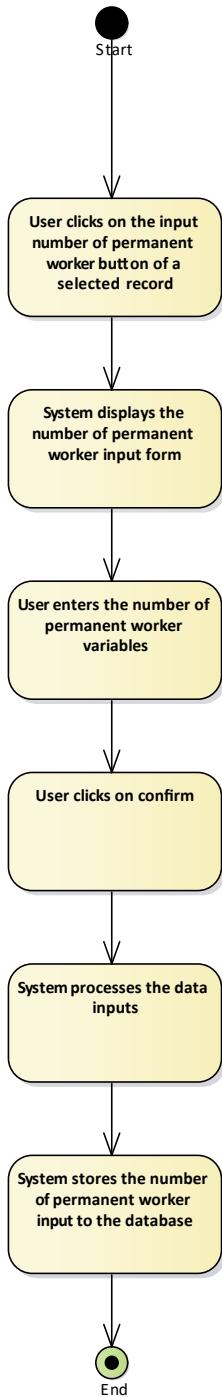


Figure 16: Input Number of Permanent Worker_ActivityGraph

System displays the number of permanent worker input form

Activity owned by 'Input Number of Permanent Worker_ActivityGraph', in package 'Use Case Model'

System displays the number of permanent worker input form
 Version 1.0 Phase 1.0 Proposed
 KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from System displays the number of permanent worker input form to User enters the number of permanent worker variables

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User clicks on the input number of permanent worker button of a selected record to System displays the number of permanent worker input form

System processes the data inputs

Activity owned by 'Input Number of Permanent Worker_ActivityGraph', in package 'Use Case Model'

System processes the data inputs
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from System processes the data inputs to System stores the number of permanent worker input to the database

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User clicks on confirm to System processes the data inputs

System stores the number of permanent worker input to the database

Activity owned by 'Input Number of Permanent Worker_ActivityGraph', in package 'Use Case Model'

System stores the number of permanent worker input to the database
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from System stores the number of permanent worker input to the database to End

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from System processes the data inputs to System stores the number of permanent worker input to the database

User clicks on confirm

Activity owned by 'Input Number of Permanent Worker_ActivityGraph', in package 'Use Case Model'

User clicks on confirm

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from User clicks on confirm to System processes the data inputs

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User enters the number of permanent worker variables to User clicks on confirm

User clicks on the input number of permanent worker button of a selected record

Activity owned by 'Input Number of Permanent Worker_ActivityGraph', in package 'Use Case Model'

User clicks on the input number of permanent worker button of a selected record

Version 1.0 Phase 1.0 Proposed

KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from User clicks on the input number of permanent worker button of a selected record to System displays the number of permanent worker input form

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from Start to User clicks on the input number of permanent worker button of a selected record

User enters the number of permanent worker variables

Activity owned by 'Input Number of Permanent Worker_ActivityGraph', in package 'Use Case Model'

User enters the number of permanent worker variables

Version 1.0 Phase 1.0 Proposed

KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from User enters the number of permanent worker variables to User clicks on confirm

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from System displays the number of permanent worker input form to User enters the number of permanent worker variables

End

ActivityFinal owned by 'Input Number of Permanent Worker_ActivityGraph', in package 'Use Case Model'

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from System stores the number of permanent worker input to the database to End

Start

ActivityInitial owned by 'Input Number of Permanent Worker_ActivityGraph', in package 'Use Case Model'

OUTGOING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from Start to User clicks on the input number of permanent worker button of a selected record

Input Number of Permanent Worker_SequenceDiagram

Interaction owned by 'Input Number of Permanent Worker', in package 'Use Case Model'

Input Number of Permanent Worker_SequenceDiagram
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

ELEMENTS OWNED BY Input Number of Permanent Worker_SequenceDiagram

█ HistoryPage : Sequence «boundary»

█ MainController : Sequence «control»

█ NumPermanentInputPage : Sequence «boundary»

█ PlanDatabase : Sequence «entity»

█ User : Sequence

1_Basic_Path diagram

Interaction diagram in package 'Use Case Model'

1_Basic_Path
Version 1.0
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

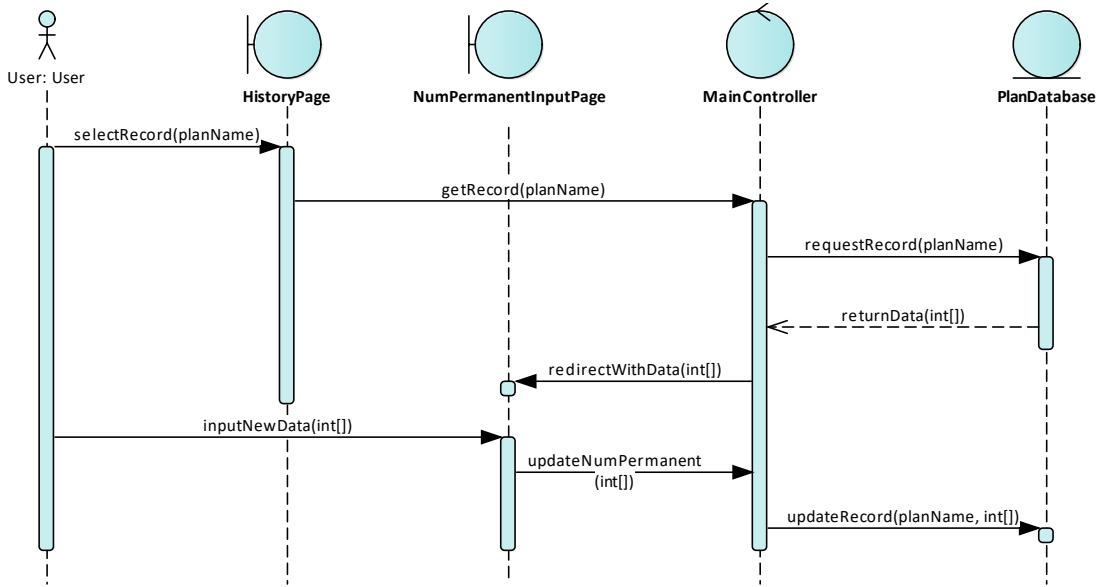


Figure 17: 1_Basic_Path

INTERACTION MESSAGES

✉ 1.0 'selectRecord(planName)' from 'User: User' sent to 'HistoryPage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.1 'getRecord' from 'HistoryPage' sent to 'MainController'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.2 'requestRecord' from 'MainController' sent to 'PlanDatabase'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.3 'returnData' from 'PlanDatabase' sent to 'MainController'.

Synchronous Call. Returns void.

[Return is True. Iteration is False. New group is False.]

✉ 1.4 'redirectWithData' from 'MainController' sent to 'NumPermanentInputPage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.5 'inputNewData' from 'User: User' sent to 'NumPermanentInputPage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.6 'updateNumPermanent' from 'NumPermanentInputPage' sent to 'MainController'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.7 'updateRecord' from 'MainController' sent to 'PlanDatabase'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

HistoryPage

Sequence «boundary» owned by 'Input Number of Permanent Worker_SequenceDiagram', in package 'Use Case Model'

HistoryPage

Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: getRecord

Sequence from «boundary» HistoryPage to «control» MainController

INCOMING BEHAVIORAL RELATIONSHIPS

Name: selectRecord(planName)

Sequence from User to «boundary» HistoryPage

MainController

Sequence «control» owned by 'Input Number of Permanent Worker_SequenceDiagram', in package 'Use Case Model'

MainController

Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: updateRecord

Sequence from «control» MainController to «entity» PlanDatabase

Name: requestRecord

Sequence from «control» MainController to «entity» PlanDatabase

Name: redirectWithData

Sequence from «control» MainController to «boundary» NumPermanentInputPage

INCOMING BEHAVIORAL RELATIONSHIPS

INCOMING BEHAVIORAL RELATIONSHIPS

Name: getRecord

➡ Sequence from «boundary» HistoryPage to «control» MainController

Name: updateNumPermanent

➡ Sequence from «boundary» NumPermanentInputPage to «control» MainController

Name: returnData

➡ Sequence from «entity» PlanDatabase to «control» MainController

NumPermanentInputPage*Sequence «boundary» owned by 'Input Number of Permanent Worker_SequenceDiagram', in package 'Use Case Model'*

NumPermanentInputPage
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: updateNumPermanent

⬅ Sequence from «boundary» NumPermanentInputPage to «control» MainController

INCOMING BEHAVIORAL RELATIONSHIPS

Name: redirectWithData

➡ Sequence from «control» MainController to «boundary» NumPermanentInputPage

Name: inputNewData

➡ Sequence from User to «boundary» NumPermanentInputPage

PlanDatabase*Sequence «entity» owned by 'Input Number of Permanent Worker_SequenceDiagram', in package 'Use Case Model'*

PlanDatabase
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: returnData

⬅ Sequence from «entity» PlanDatabase to «control» MainController

INCOMING BEHAVIORAL RELATIONSHIPS

Name: updateRecord

➡ Sequence from «control» MainController to «entity» PlanDatabase

INCOMING BEHAVIORAL RELATIONSHIPS

Name: requestRecord
 Sequence from «control» MainController to «entity» PlanDatabase

User

Sequence owned by 'Input Number of Permanent Worker_SequenceDiagram', in package 'Use Case Model'

User
 Version 1.0 Phase 1.0 Proposed
 KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: selectRecord(planName)
 Sequence from User to «boundary» HistoryPage

Name: inputNewData
 Sequence from User to «boundary» NumPermanentInputPage

Input Permanent Worker Production

UseCase in package 'Use Case Model'

Input Permanent Worker Production
 Version 1.0 Phase 1.0 Proposed
 ASUS created on 29-Jan-23. Last modified 29-Jan-23

ELEMENTS OWNED BY Input Permanent Worker Production

-  Input Permanent Worker Production_ActivityGraph : Activity
-  Input Permanent Worker Production_SequenceDiagram : Interaction

SCENARIOS

-  Basic Path. Basic Path
1. User clicks on the input permanent worker production button of a selected record
 2. System displays the permanent worker production input form
 3. User enters the permanent worker production variables
 4. User clicks on confirm
 5. System processes the data inputs
 6. System stores the permanent worker production input to the database

CONSTRAINTS

CONSTRAINTS	
⌚ Pre-condition. The user is logged in to the system	[Mandatory, Weight is 0.]
⌚ Post-condition. The permanent worker production number has been inputted/modified	[Mandatory, Weight is 1.]

CONNECTORS	
 Extend «extend» Source -> Destination From: View Optimized Plan : UseCase, Public To: Input Permanent Worker Production : UseCase, Public	

Input Permanent Worker Production_ActivityGraph

Activity owned by 'Input Permanent Worker Production', in package 'Use Case Model'

Input Permanent Worker Production_ActivityGraph
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

ELEMENTS OWNED BY Input Permanent Worker Production_ActivityGraph	
▣ System displays the permanent worker production input form : Activity	
▣ System processes the data inputs : Activity	
▣ System stores the permanent worker production input to the database : Activity	
▣ User clicks on confirm : Activity	
▣ User clicks on the input permanent worker production button of a selected record : Activity	
▣ User enters the permanent worker production variables : Activity	
▣ End : ActivityFinal	
▣ Start : ActivityInitial	

Input Permanent Worker Production_ActivityGraph diagram

Activity diagram in package 'Use Case Model'

Input Permanent Worker Production_ActivityGraph

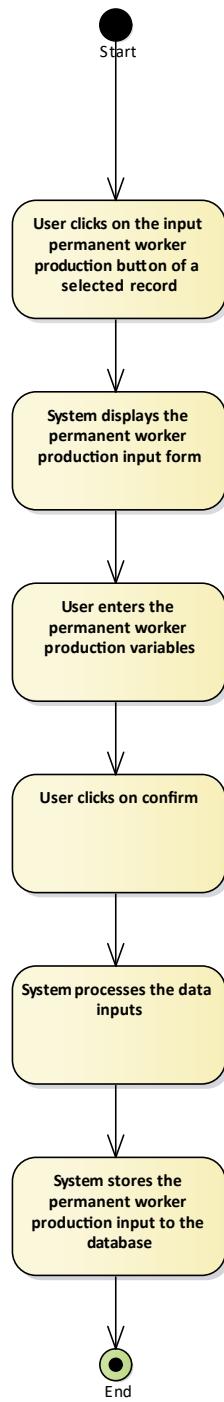


Figure 18: Input Permanent Worker Production_ActivityGraph

System displays the permanent worker production input form

Activity owned by 'Input Permanent Worker Production_ActivityGraph', in package 'Use Case Model'

System displays the permanent worker production input form

Version 1.0 Phase 1.0 Proposed

KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from System displays the permanent worker production input form to User enters the permanent worker production variables

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User clicks on the input permanent worker production button of a selected record to System displays the permanent worker production input form

System processes the data inputs

Activity owned by 'Input Permanent Worker Production_ActivityGraph', in package 'Use Case Model'

System processes the data inputs
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from System processes the data inputs to System stores the permanent worker production input to the database

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User clicks on confirm to System processes the data inputs

System stores the permanent worker production input to the database

Activity owned by 'Input Permanent Worker Production_ActivityGraph', in package 'Use Case Model'

System stores the permanent worker production input to the database
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from System stores the permanent worker production input to the database to End

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from System processes the data inputs to System stores the permanent worker production input to the database

User clicks on confirm

Activity owned by 'Input Permanent Worker Production_ActivityGraph', in package 'Use Case Model'

User clicks on confirm
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from User clicks on confirm to System processes the data inputs

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User enters the permanent worker production variables to User clicks on confirm

User clicks on the input permanent worker production button of a selected record

Activity owned by 'Input Permanent Worker Production_ActivityGraph', in package 'Use Case Model'

User clicks on the input permanent worker production button of a selected record
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from User clicks on the input permanent worker production button of a selected record to System displays the permanent worker production input form

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from Start to User clicks on the input permanent worker production button of a selected record

User enters the permanent worker production variables

Activity owned by 'Input Permanent Worker Production_ActivityGraph', in package 'Use Case Model'

User enters the permanent worker production variables
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from User enters the permanent worker production variables to User clicks on confirm

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from System displays the permanent worker production input form to User enters the permanent worker production variables

End

ActivityFinal owned by 'Input Permanent Worker Production_ActivityGraph', in package 'Use Case Model'

INCOMING BEHAVIORAL RELATIONSHIPS

- ⇒ ControlFlow from System stores the permanent worker production input to the database to End

Start

ActivityInitial owned by 'Input Permanent Worker Production_ActivityGraph', in package 'Use Case Model'

OUTGOING BEHAVIORAL RELATIONSHIPS

- ⇒ ControlFlow from Start to User clicks on the input permanent worker production button of a selected record

Input Permanent Worker Production_SequenceDiagram

Interaction owned by 'Input Permanent Worker Production', in package 'Use Case Model'

Input Permanent Worker Production_SequenceDiagram
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

ELEMENTS OWNED BY Input Permanent Worker Production_SequenceDiagram

- █ HistoryPage : Sequence «boundary»

- █ MainController : Sequence «control»

- █ PermanentProdPage : Sequence «boundary»

- █ PlanDatabase : Sequence «entity»

- █ User : Sequence

1_Basic_Path diagram

Interaction diagram in package 'Use Case Model'

1_Basic_Path
Version 1.0
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

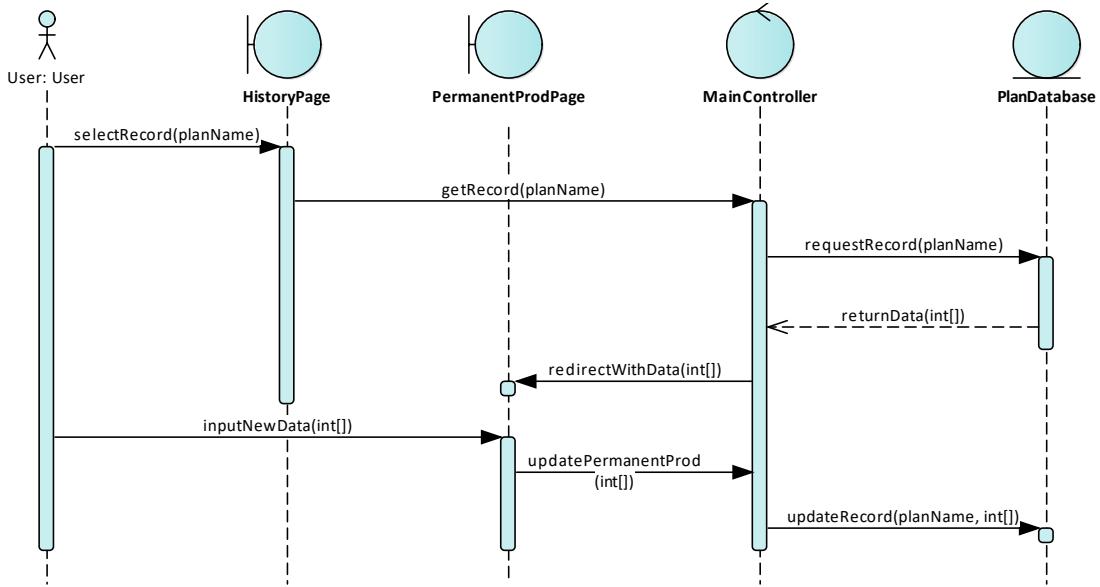


Figure 19: 1_Basic_Path

INTERACTION MESSAGES

- ✉ 1.0 'selectRecord(planName)' from 'User: User' sent to 'HistoryPage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

- ✉ 1.1 'getRecord' from 'HistoryPage' sent to 'MainController'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

- ✉ 1.2 'requestRecord' from 'MainController' sent to 'PlanDatabase'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

- ✉ 1.3 'returnData' from 'PlanDatabase' sent to 'MainController'.

Synchronous Call. Returns void.

[Return is True. Iteration is False. New group is False.]

- ✉ 1.4 'redirectWithData' from 'MainController' sent to 'PermanentProdPage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

- ✉ 1.5 'inputNewData' from 'User: User' sent to 'PermanentProdPage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.6 'updatePermanentProd' from 'PermanentProdPage' sent to 'MainController'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.7 'updateRecord' from 'MainController' sent to 'PlanDatabase'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

HistoryPage

Sequence «boundary» owned by 'Input Permanent Worker Production_SequenceDiagram', in package 'Use Case Model'

HistoryPage

Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: getRecord

Sequence from «boundary» HistoryPage to «control» MainController

INCOMING BEHAVIORAL RELATIONSHIPS

Name: selectRecord(planName)

Sequence from User to «boundary» HistoryPage

MainController

Sequence «control» owned by 'Input Permanent Worker Production_SequenceDiagram', in package 'Use Case Model'

MainController

Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: redirectWithData

Sequence from «control» MainController to «boundary» PermanentProdPage

Name: requestRecord

Sequence from «control» MainController to «entity» PlanDatabase

Name: updateRecord

Sequence from «control» MainController to «entity» PlanDatabase

INCOMING BEHAVIORAL RELATIONSHIPS

INCOMING BEHAVIORAL RELATIONSHIPS

Name: updatePermanentProd

➡ Sequence from «boundary» PermanentProdPage to «control» MainController

Name: returnData

➡ Sequence from «entity» PlanDatabase to «control» MainController

Name: getRecord

➡ Sequence from «boundary» HistoryPage to «control» MainController

PermanentProdPage

Sequence «boundary» owned by 'Input Permanent Worker Production_SequenceDiagram', in package 'Use Case Model'

PermanentProdPage

Version 1.0 Phase 1.0 Proposed

KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: updatePermanentProd

⬅ Sequence from «boundary» PermanentProdPage to «control» MainController

INCOMING BEHAVIORAL RELATIONSHIPS

Name: redirectWithData

➡ Sequence from «control» MainController to «boundary» PermanentProdPage

Name: inputNewData

➡ Sequence from User to «boundary» PermanentProdPage

PlanDatabase

Sequence «entity» owned by 'Input Permanent Worker Production_SequenceDiagram', in package 'Use Case Model'

PlanDatabase

Version 1.0 Phase 1.0 Proposed

KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: returnData

⬅ Sequence from «entity» PlanDatabase to «control» MainController

INCOMING BEHAVIORAL RELATIONSHIPS

Name: requestRecord

➡ Sequence from «control» MainController to «entity» PlanDatabase

INCOMING BEHAVIORAL RELATIONSHIPS

Name: updateRecord
 Sequence from «control» MainController to «entity» PlanDatabase

User

Sequence owned by 'Input Permanent Worker Production_SequenceDiagram', in package 'Use Case Model'

User
 Version 1.0 Phase 1.0 Proposed
 KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: selectRecord(planName)
 Sequence from User to «boundary» HistoryPage

Name: inputNewData
 Sequence from User to «boundary» PermanentProdPage

Input Starting and Ending Inventory Level

UseCase in package 'Use Case Model'

Input Starting and Ending Inventory Level
 Version 1.0 Phase 1.0 Proposed
 muhiq created on 13-Nov-22. Last modified 29-Jan-23

ELEMENTS OWNED BY Input Starting and Ending Inventory Level

-  Input Starting and Ending Inventory Level_ActivityGraph : Activity
-  Input Starting and Ending Inventory Level_SequenceDiagram : Interaction

SCENARIOS

-  Basic Path. Basic Path
 1. User clicks on the input starting and ending inventory number of a selected record
 2. System displays the starting and ending inventory input form
 3. User enters the starting and ending inventory variables
 4. User clicks on confirm
 5. System processes the data inputs
 6. System stores the starting and ending inventory input to the database

-  Alternate. Number of previous month's temporary workers not exist
 1. Print Message to enter the details of previous month

SCENARIOS
<p>Alternate. Number of temporary workers hired and fired not exist</p> <p>1. <include> Calculate firing and hiring cost</p>

CONSTRAINTS
<p>Pre-condition. The user is logged in to the system [Mandatory, Weight is 0.]</p> <p>Post-condition. The starting and ending inventory number has been inputted/modified [Mandatory, Weight is 1.]</p>

CONNECTORS
<p> Extend «extend» Source -> Destination From: View Optimized Plan : UseCase, Public To: Input Starting and Ending Inventory Level : UseCase, Public</p>

Input Starting and Ending Inventory Level_ActivityGraph

Activity owned by 'Input Starting and Ending Inventory Level', in package 'Use Case Model'

Input Starting and Ending Inventory Level_ActivityGraph
 Version 1.0 Phase 1.0 Proposed
 KingZexx created on 29-Jan-23. Last modified 29-Jan-23

ELEMENTS OWNED BY Input Starting and Ending Inventory Level_ActivityGraph
System displays the starting and ending inventory input form : Activity
System processes the data inputs : Activity
System stores the starting and ending inventory input to the database : Activity
User clicks on confirm : Activity
User clicks on the input starting and ending inventory number of a selected record : Activity
User enters the starting and ending inventory variables : Activity
End : ActivityFinal
Start : ActivityInitial

Input Starting and Ending Inventory Level_ActivityGraph diagram

Activity diagram in package 'Use Case Model'

Input Starting and Ending Inventory Level_ActivityGraph
Version 1.0
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

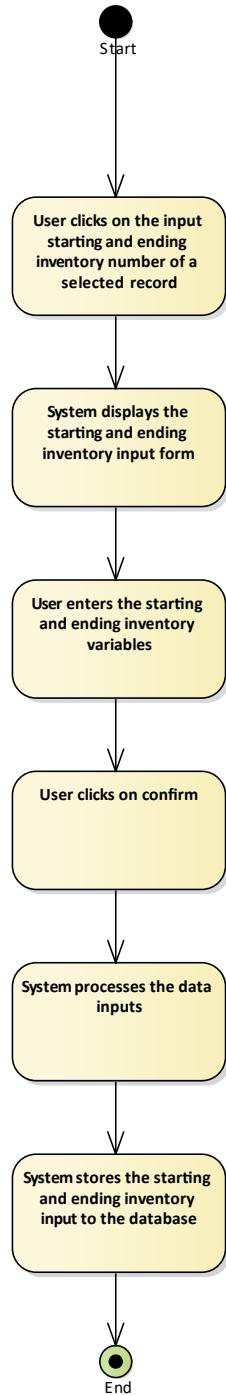


Figure 20: Input Starting and Ending Inventory Level_ActivityGraph

System displays the starting and ending inventory input form

Activity owned by 'Input Starting and Ending Inventory Level_ActivityGraph', in package 'Use Case Model'

System displays the starting and ending inventory input form
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from System displays the starting and ending inventory input form to User enters the starting and ending inventory variables

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User clicks on the input starting and ending inventory number of a selected record to System displays the starting and ending inventory input form

System processes the data inputs

Activity owned by 'Input Starting and Ending Inventory Level_ActivityGraph', in package 'Use Case Model'

System processes the data inputs
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from System processes the data inputs to System stores the starting and ending inventory input to the database

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User clicks on confirm to System processes the data inputs

System stores the starting and ending inventory input to the database

Activity owned by 'Input Starting and Ending Inventory Level_ActivityGraph', in package 'Use Case Model'

System stores the starting and ending inventory input to the database
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from System stores the starting and ending inventory input to the database to End

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from System processes the data inputs to System stores the starting and ending inventory input to the database

INCOMING BEHAVIORAL RELATIONSHIPS**User clicks on confirm***Activity owned by 'Input Starting and Ending Inventory Level_ActivityGraph', in package 'Use Case Model'*

User clicks on confirm
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from User clicks on confirm to System processes the data inputs

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User enters the starting and ending inventory variables to User clicks on confirm

User clicks on the input starting and ending inventory number of a selected record*Activity owned by 'Input Starting and Ending Inventory Level_ActivityGraph', in package 'Use Case Model'*

User clicks on the input starting and ending inventory number of a selected record
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from User clicks on the input starting and ending inventory number of a selected record to System displays the starting and ending inventory input form

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from Start to User clicks on the input starting and ending inventory number of a selected record

User enters the starting and ending inventory variables*Activity owned by 'Input Starting and Ending Inventory Level_ActivityGraph', in package 'Use Case Model'*

User enters the starting and ending inventory variables
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from User enters the starting and ending inventory variables to User clicks on confirm

INCOMING BEHAVIORAL RELATIONSHIPS

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from System displays the starting and ending inventory input form to User enters the starting and ending inventory variables

End

ActivityFinal owned by 'Input Starting and Ending Inventory Level_ActivityGraph', in package 'Use Case Model'

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from System stores the starting and ending inventory input to the database to End

Start

ActivityInitial owned by 'Input Starting and Ending Inventory Level_ActivityGraph', in package 'Use Case Model'

OUTGOING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from Start to User clicks on the input starting and ending inventory number of a selected record

Input Starting and Ending Inventory Level_SequenceDiagram

Interaction owned by 'Input Starting and Ending Inventory Level', in package 'Use Case Model'

Input Starting and Ending Inventory Level_SequenceDiagram
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

ELEMENTS OWNED BY Input Starting and Ending Inventory Level_SequenceDiagram

█ HistoryPage : Sequence «boundary»

█ MainController : Sequence «control»

█ PlanDatabase : Sequence «entity»

█ StartEndInventoryInputPage : Sequence «boundary»

█ User : Sequence

1_Basic_Path diagram

Interaction diagram in package 'Use Case Model'

1_Basic_Path
Version 1.0
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

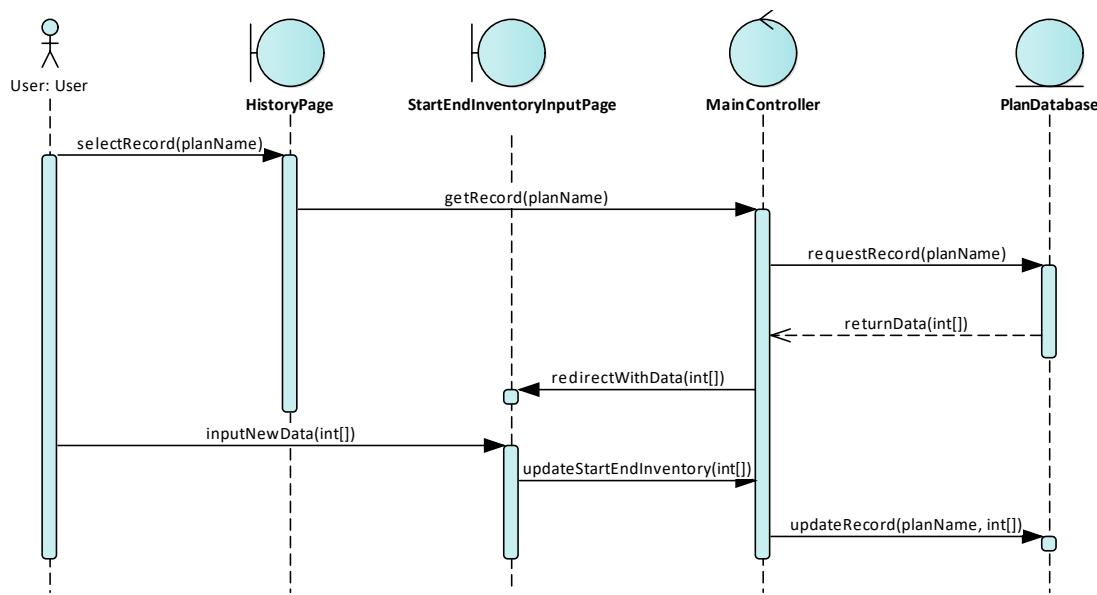


Figure 21: 1_Basic_Path

INTERACTION MESSAGES

- ✉ 1.0 'selectRecord(planName)' from 'User: User' sent to 'HistoryPage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

- ✉ 1.1 'getRecord' from 'HistoryPage' sent to 'MainController'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

- ✉ 1.2 'requestRecord' from 'MainController' sent to 'PlanDatabase'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

- ✉ 1.3 'returnData' from 'PlanDatabase' sent to 'MainController'.

Synchronous Call. Returns void.

[Return is True. Iteration is False. New group is False.]

- ✉ 1.4 'redirectWithData' from 'MainController' sent to 'StartEndInventoryInputPage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.5 'inputNewData' from 'User: User' sent to 'StartEndInventoryInputPage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.6 'updateStartEndInventory' from 'StartEndInventoryInputPage' sent to 'MainController'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.7 'updateRecord' from 'MainController' sent to 'PlanDatabase'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

HistoryPage

Sequence «boundary» owned by 'Input Starting and Ending Inventory Level_SequenceDiagram', in package 'Use Case Model'

HistoryPage

Version 1.0 Phase 1.0 Proposed

KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: `getRecord`

Sequence from «boundary» HistoryPage to «control» MainController

INCOMING BEHAVIORAL RELATIONSHIPS

Name: `selectRecord(planName)`

Sequence from User to «boundary» HistoryPage

MainController

Sequence «control» owned by 'Input Starting and Ending Inventory Level_SequenceDiagram', in package 'Use Case Model'

MainController

Version 1.0 Phase 1.0 Proposed

KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: `redirectWithData`

Sequence from «control» MainController to «boundary» StartEndInventoryInputPage

Name: `requestRecord`

Sequence from «control» MainController to «entity» PlanDatabase

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: updateRecord
 ↳ Sequence from «control» MainController to «entity» PlanDatabase

INCOMING BEHAVIORAL RELATIONSHIPS

Name: returnData
 ⇢ Sequence from «entity» PlanDatabase to «control» MainController

Name: updateStartEndInventory
 ⇢ Sequence from «boundary» StartEndInventoryInputPage to «control» MainController

Name: getRecord
 ⇢ Sequence from «boundary» HistoryPage to «control» MainController

PlanDatabase

Sequence «entity» owned by 'Input Starting and Ending Inventory Level_SequenceDiagram', in package 'Use Case Model'

PlanDatabase
 Version 1.0 Phase 1.0 Proposed
 KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: returnData
 ↳ Sequence from «entity» PlanDatabase to «control» MainController

INCOMING BEHAVIORAL RELATIONSHIPS

Name: requestRecord
 ⇢ Sequence from «control» MainController to «entity» PlanDatabase

Name: updateRecord
 ⇢ Sequence from «control» MainController to «entity» PlanDatabase

StartEndInventoryInputPage

Sequence «boundary» owned by 'Input Starting and Ending Inventory Level_SequenceDiagram', in package 'Use Case Model'

StartEndInventoryInputPage
 Version 1.0 Phase 1.0 Proposed
 KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: updateStartEndInventory
 ↳ Sequence from «boundary» StartEndInventoryInputPage to «control» MainController

INCOMING BEHAVIORAL RELATIONSHIPS

Name: inputData
 ↲ Sequence from User to «boundary» StartEndInventoryInputPage

Name: redirectWithData
 ↲ Sequence from «control» MainController to «boundary» StartEndInventoryInputPage

User

Sequence owned by 'Input Starting and Ending Inventory Level_SequenceDiagram', in package 'Use Case Model'

User
 Version 1.0 Phase 1.0 Proposed
 KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: inputData
 ↳ Sequence from User to «boundary» StartEndInventoryInputPage

Name: selectRecord(planName)
 ↳ Sequence from User to «boundary» HistoryPage

Input Temporary Worker Production

UseCase in package 'Use Case Model'

Input Temporary Worker Production
 Version 1.0 Phase 1.0 Proposed
 ASUS created on 29-Jan-23. Last modified 29-Jan-23

ELEMENTS OWNED BY Input Temporary Worker Production

█ Input Temporary Worker Production_ActivityGraph : Activity

█ Input Temporary Worker Production_SequenceDiagram : Interaction

SCENARIOS

█ Basic Path. Basic Path

1. User clicks on the input temporary worker production button of a selected record
2. System displays the temporary worker production input form

SCENARIOS

3. User enters temporary worker production variables
4. User clicks on confirm
5. System processes the data inputs
6. System stores the temporary worker production input to the database

CONSTRAINTS

-  Pre-condition. The user is logged in to the system [Mandatory, Weight is 0.]
-  Post-condition. The temporary worker production number has been inputted/modified [Approved, Weight is 1.]

CONNECTORS

-  **Extend** «extend» Source -> Destination
 From: View Optimized Plan : UseCase, Public
 To: Input Temporary Worker Production : UseCase, Public

Input Temporary Worker Production_ActivityGraph

Activity owned by 'Input Temporary Worker Production', in package 'Use Case Model'

Input Temporary Worker Production_ActivityGraph
 Version 1.0 Phase 1.0 Proposed
 KingZexx created on 29-Jan-23. Last modified 29-Jan-23

ELEMENTS OWNED BY Input Temporary Worker Production_ActivityGraph

-  System displays the temporary worker production input form : Activity
-  System processes the data inputs : Activity
-  System stores the temporary worker production input to the database : Activity
-  User clicks on confirm : Activity
-  User clicks on the input temporary worker production button of a selected record : Activity
-  User enters temporary worker production variables : Activity
-  End : ActivityFinal
-  Start : ActivityInitial

Input Temporary Worker Production_ActivityGraph diagram

Activity diagram in package 'Use Case Model'

Input Temporary Worker Production_ActivityGraph
Version 1.0
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

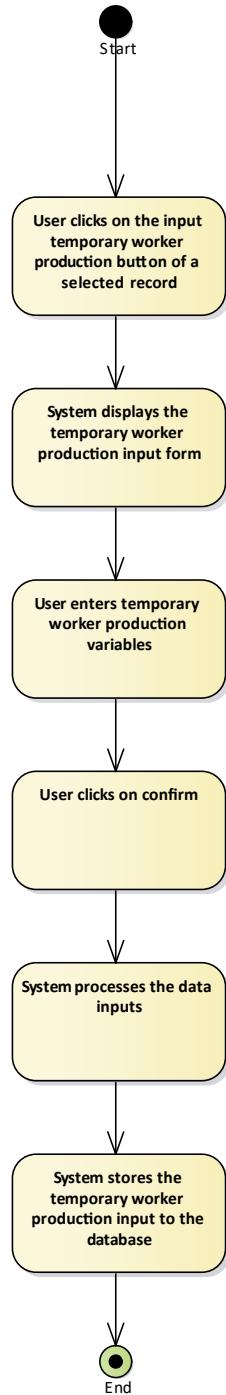


Figure 22: Input Temporary Worker Production_ActivityGraph

System displays the temporary worker production input form

Activity owned by 'Input Temporary Worker Production_ActivityGraph', in package 'Use Case Model'

System displays the temporary worker production input form
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from System displays the temporary worker production input form to User enters temporary worker production variables

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User clicks on the input temporary worker production button of a selected record to System displays the temporary worker production input form

System processes the data inputs

Activity owned by 'Input Temporary Worker Production_ActivityGraph', in package 'Use Case Model'

System processes the data inputs
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from System processes the data inputs to System stores the temporary worker production input to the database

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User clicks on confirm to System processes the data inputs

System stores the temporary worker production input to the database

Activity owned by 'Input Temporary Worker Production_ActivityGraph', in package 'Use Case Model'

System stores the temporary worker production input to the database
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from System stores the temporary worker production input to the database to End

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from System processes the data inputs to System stores the temporary worker production input to the database

INCOMING BEHAVIORAL RELATIONSHIPS**User clicks on confirm***Activity owned by 'Input Temporary Worker Production_ActivityGraph', in package 'Use Case Model'*

User clicks on confirm
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from User clicks on confirm to System processes the data inputs

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User enters temporary worker production variables to User clicks on confirm

User clicks on the input temporary worker production button of a selected record*Activity owned by 'Input Temporary Worker Production_ActivityGraph', in package 'Use Case Model'*

User clicks on the input temporary worker production button of a selected record
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from User clicks on the input temporary worker production button of a selected record to System displays the temporary worker production input form

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from Start to User clicks on the input temporary worker production button of a selected record

User enters temporary worker production variables*Activity owned by 'Input Temporary Worker Production_ActivityGraph', in package 'Use Case Model'*

User enters temporary worker production variables
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from User enters temporary worker production variables to User clicks on confirm

INCOMING BEHAVIORAL RELATIONSHIPS

INCOMING BEHAVIORAL RELATIONSHIPS

- ⇒ ControlFlow from System displays the temporary worker production input form to User enters temporary worker production variables

End

ActivityFinal owned by 'Input Temporary Worker Production_ActivityGraph', in package 'Use Case Model'

INCOMING BEHAVIORAL RELATIONSHIPS

- ⇒ ControlFlow from System stores the temporary worker production input to the database to End

Start

ActivityInitial owned by 'Input Temporary Worker Production_ActivityGraph', in package 'Use Case Model'

OUTGOING BEHAVIORAL RELATIONSHIPS

- ⇒ ControlFlow from Start to User clicks on the input temporary worker production button of a selected record

Input Temporary Worker Production_SequenceDiagram

Interaction owned by 'Input Temporary Worker Production', in package 'Use Case Model'

Input Temporary Worker Production_SequenceDiagram
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

ELEMENTS OWNED BY Input Temporary Worker Production_SequenceDiagram

█ HistoryPage : Sequence «boundary»

█ MainController : Sequence «control»

█ PlanDatabase : Sequence «entity»

█ TemporaryProdPage : Sequence «boundary»

█ User : Sequence

1_Basic_Path diagram

Interaction diagram in package 'Use Case Model'

1_Basic_Path
Version 1.0
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

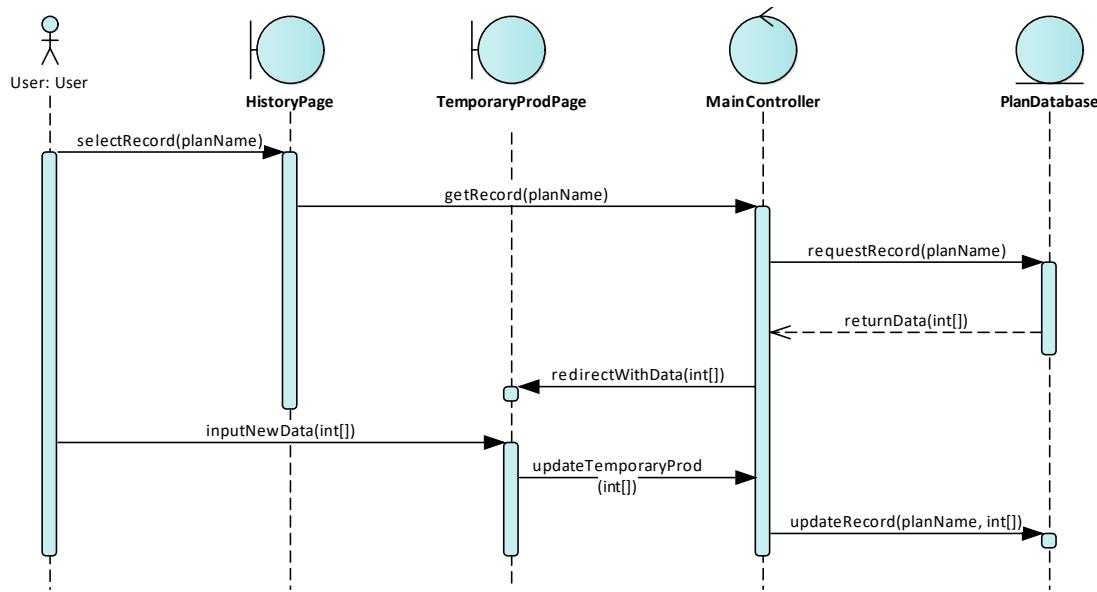


Figure 23: 1_Basic_Path

INTERACTION MESSAGES

- ✉ 1.0 'selectRecord(planName)' from 'User: User' sent to 'HistoryPage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

- ✉ 1.1 'getRecord' from 'HistoryPage' sent to 'MainController'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

- ✉ 1.2 'requestRecord' from 'MainController' sent to 'PlanDatabase'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

- ✉ 1.3 'returnData' from 'PlanDatabase' sent to 'MainController'.

Synchronous Call. Returns void.

[Return is True. Iteration is False. New group is False.]

- ✉ 1.4 'redirectWithData' from 'MainController' sent to 'TemporaryProdPage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.5 'inputNewData' from 'User: User' sent to 'TemporaryProdPage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.6 'updateTemporaryProd' from 'TemporaryProdPage' sent to 'MainController'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.7 'updateRecord' from 'MainController' sent to 'PlanDatabase'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

HistoryPage

Sequence «boundary» owned by 'Input Temporary Worker Production_SequenceDiagram', in package 'Use Case Model'

HistoryPage

Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: `getRecord`

Sequence from «boundary» HistoryPage to «control» MainController

INCOMING BEHAVIORAL RELATIONSHIPS

Name: `selectRecord(planName)`

Sequence from User to «boundary» HistoryPage

MainController

Sequence «control» owned by 'Input Temporary Worker Production_SequenceDiagram', in package 'Use Case Model'

MainController

Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: `updateRecord`

Sequence from «control» MainController to «entity» PlanDatabase

Name: `redirectWithData`

Sequence from «control» MainController to «boundary» TemporaryProdPage

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: requestRecord
 ↳ Sequence from «control» MainController to «entity» PlanDatabase

INCOMING BEHAVIORAL RELATIONSHIPS

Name: getRecord
 ⇢ Sequence from «boundary» HistoryPage to «control» MainController

Name: updateTemporaryProd
 ⇢ Sequence from «boundary» TemporaryProdPage to «control» MainController

Name: returnData
 ⇢ Sequence from «entity» PlanDatabase to «control» MainController

PlanDatabase

Sequence «entity» owned by 'Input Temporary Worker Production_SequenceDiagram', in package 'Use Case Model'

PlanDatabase
 Version 1.0 Phase 1.0 Proposed
 KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: returnData
 ↳ Sequence from «entity» PlanDatabase to «control» MainController

INCOMING BEHAVIORAL RELATIONSHIPS

Name: updateRecord
 ⇢ Sequence from «control» MainController to «entity» PlanDatabase

Name: requestRecord
 ⇢ Sequence from «control» MainController to «entity» PlanDatabase

TemporaryProdPage

Sequence «boundary» owned by 'Input Temporary Worker Production_SequenceDiagram', in package 'Use Case Model'

TemporaryProdPage
 Version 1.0 Phase 1.0 Proposed
 KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: updateTemporaryProd
 ⇢ Sequence from «boundary» TemporaryProdPage to «control» MainController

INCOMING BEHAVIORAL RELATIONSHIPS

Name: redirectWithData

➡ Sequence from «control» MainController to «boundary» TemporaryProdPage

Name: inputData

➡ Sequence from User to «boundary» TemporaryProdPage

User

Sequence owned by 'Input Temporary Worker Production_SequenceDiagram', in package 'Use Case Model'

User

Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: selectRecord(planName)

⬅ Sequence from User to «boundary» HistoryPage

Name: inputData

⬅ Sequence from User to «boundary» TemporaryProdPage

Login

UseCase in package 'Use Case Model'

Login

Version 1.0 Phase 1.0 Proposed
ASUS created on 22-Nov-22. Last modified 29-Jan-23

ELEMENTS OWNED BY Login

█ Login_ActivityGraphWithParam : Activity

SCENARIOS

█ Basic Path. Basic Path

1. User enters login name and password
2. User click "Login" button
3. System verify the details
Exception: 3a. If the details are incorrect
4. User successfully logs in

█ Exception. If the details are incorrect

1. System display "Incorrect username or password"

SCENARIOS
CONSTRAINTS
<p>⌚ Pre-condition. User have registered account. [Approved, Weight is 0.]</p> <p>⌚ Post-condition. User has successfully logs into the system [Approved, Weight is 1.]</p>
CONNECTORS
 UseCaseLink Source -> Destination From: User : Actor, Public To: Login : UseCase, Public

Login_ActivityGraphWithParam

Activity owned by 'Login', in package 'Use Case Model'

Login_ActivityGraphWithParam
 Version 1.0 Phase 1.0 Proposed
 NITRO created on 29-Jan-23. Last modified 29-Jan-23

ELEMENTS OWNED BY Login_ActivityGraphWithParam
█ System display "Incorrect username or password" : Activity
█ System verify the details : Activity
█ User click "Login" button : Activity
█ User enters login name and password : Activity
█ User successfully logs in : Activity
█ Exception1 : DecisionNode
█ End : ActivityFinal
█ Start : ActivityInitial

Login_ActivityGraphWithParam diagram

Activity diagram in package 'Use Case Model'

Login_ActivityGraphWithParam

Version 1.0

NITRO created on 29-Jan-23. Last modified 29-Jan-23

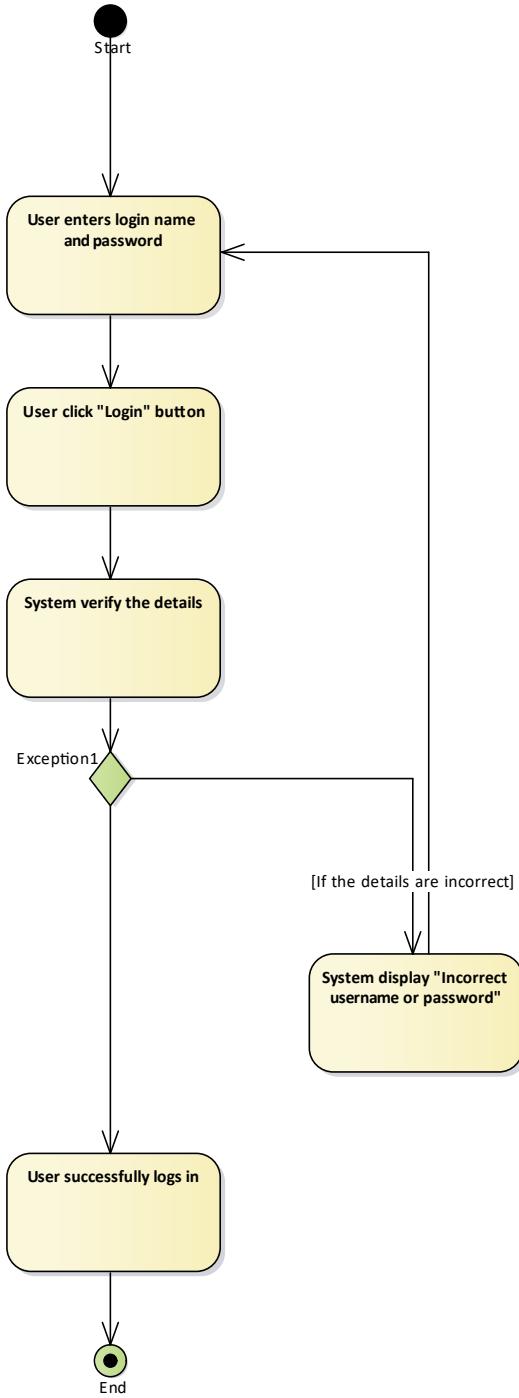


Figure 24: Login_ActivityGraphWithParam

System display "Incorrect username or password"*Activity owned by 'Login_ActivityGraphWithParam', in package 'Use Case Model'*

System display "Incorrect username or password"

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from System display "Incorrect username or password" to User enters login name and password

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from Exception1 to System display "Incorrect username or password"
Guard: If the details are incorrect

System verify the details

Activity owned by 'Login_ActivityGraphWithParam', in package 'Use Case Model'

System verify the details
Version 1.0 Phase 1.0 Proposed
NITRO created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from System verify the details to Exception1

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User click "Login" button to System verify the details

User click "Login" button

Activity owned by 'Login_ActivityGraphWithParam', in package 'Use Case Model'

User click "Login" button
Version 1.0 Phase 1.0 Proposed
NITRO created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from User click "Login" button to System verify the details

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User enters login name and password to User click "Login" button

User enters login name and password

Activity owned by 'Login_ActivityGraphWithParam', in package 'Use Case Model'

User enters login name and password
Version 1.0 Phase 1.0 Proposed
NITRO created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from User enters login name and password to User click "Login" button

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from System display "Incorrect username or password" to User enters login name and password

⇒ ControlFlow from Start to User enters login name and password

User successfully logs in

Activity owned by 'Login_ActivityGraphWithParam', in package 'Use Case Model'

User successfully logs in
Version 1.0 Phase 1.0 Proposed
NITRO created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from User successfully logs in to End

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from Exception1 to User successfully logs in

Exception1

DecisionNode owned by 'Login_ActivityGraphWithParam', in package 'Use Case Model'

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from Exception1 to System display "Incorrect username or password"
Guard: If the details are incorrect

↳ ControlFlow from Exception1 to User successfully logs in

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from System verify the details to Exception1

End

ActivityFinal owned by 'Login_ActivityGraphWithParam', in package 'Use Case Model'

INCOMING BEHAVIORAL RELATIONSHIPS

- ➡ ControlFlow from User successfully logs in to End

Start

ActivityInitial owned by 'Login_ActivityGraphWithParam', in package 'Use Case Model'

OUTGOING BEHAVIORAL RELATIONSHIPS

- ⬅ ControlFlow from Start to User enters login name and password

Logout

UseCase in package 'Use Case Model'

Logout
Version 1.0 Phase 1.0 Proposed
ASUS created on 22-Nov-22. Last modified 29-Jan-23

SCENARIOS

 Basic Path. Basic Path

1. User click logout
2. Request logout to logout controller (LC)
3. Confirmation logout to logout controller (LC)

Alternate: 3a. User changes mind
4. Redirect back to login page

 Alternate. User changes mind

1. Back to previous page

CONNECTORS

 **UseCaseLink** Source -> Destination

From: User : Actor, Public
 To: Logout : UseCase, Public

Send Feedback

UseCase in package 'Use Case Model'

Send Feedback
Version 1.0 Phase 1.0 Proposed

ELEMENTS OWNED BY Send Feedback

 Send Feedback_ActivityGraphWithParam : Activity

 Send Feedback : Interaction

SCENARIOS

 Basic Path. Basic Path

1. User click "contact"
2. User input email address
3. User input feedback subject
4. User input feedback content
5. User click "Submit" button

CONSTRAINTS

 Pre-condition. User has successfully logged into the system

[Mandatory, Weight is 0.]

 Post-condition. User has successfully submitted feedback

[Mandatory, Weight is 1.]

CONNECTORS

 **UseCaseLink** Source -> Destination

From: User : Actor, Public

To: Send Feedback : UseCase, Public

Send Feedback_ActivityGraphWithParam

Activity owned by 'Send Feedback', in package 'Use Case Model'

Send Feedback_ActivityGraphWithParam

Version 1.0 Phase 1.0 Proposed

Lenovo created on 29-Jan-23. Last modified 29-Jan-23

ELEMENTS OWNED BY Send Feedback_ActivityGraphWithParam

 User click "contact" : Activity

 User click "Submit" button : Activity

 User input email address : Activity

ELEMENTS OWNED BY Send Feedback_ActivityGraphWithParam	
█	User input feedback content : Activity
█	User input feedback subject : Activity
█	End : ActivityFinal
█	Start : ActivityInitial

Send Feedback_ActivityGraphWithParam diagram

Activity diagram in package 'Use Case Model'

Send Feedback_ActivityGraphWithParam

Version 1.0

Lenovo created on 29-Jan-23. Last modified 29-Jan-23

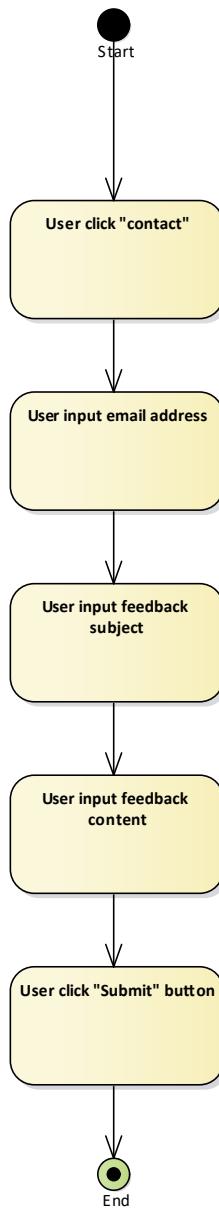


Figure 25: Send Feedback_ActivityGraphWithParam

User click "contact"

Activity owned by 'Send Feedback_ActivityGraphWithParam', in package 'Use Case Model'

User click "contact"
Version 1.0 Phase 1.0 Proposed
Lenovo created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS
↳ ControlFlow from User click "contact" to User input email address
INCOMING BEHAVIORAL RELATIONSHIPS

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from Start to User click "contact"

User click "Submit" button

Activity owned by 'Send Feedback_ActivityGraphWithParam', in package 'Use Case Model'

User click "Submit" button
Version 1.0 Phase 1.0 Proposed
Lenovo created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

⇐ ControlFlow from User click "Submit" button to End

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User input feedback content to User click "Submit" button

User input email address

Activity owned by 'Send Feedback_ActivityGraphWithParam', in package 'Use Case Model'

User input email address
Version 1.0 Phase 1.0 Proposed
Lenovo created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

⇐ ControlFlow from User input email address to User input feedback subject

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User click "contact" to User input email address

User input feedback content

Activity owned by 'Send Feedback_ActivityGraphWithParam', in package 'Use Case Model'

User input feedback content
Version 1.0 Phase 1.0 Proposed
Lenovo created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

⇐ ControlFlow from User input feedback content to User click "Submit" button

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User input feedback subject to User input feedback content

User input feedback subject

Activity owned by 'Send Feedback_ActivityGraphWithParam', in package 'Use Case Model'

User input feedback subject
Version 1.0 Phase 1.0 Proposed
Lenovo created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

⇐ ControlFlow from User input feedback subject to User input feedback content

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User input email address to User input feedback subject

End

ActivityFinal owned by 'Send Feedback_ActivityGraphWithParam', in package 'Use Case Model'

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User click "Submit" button to End

Start

ActivityInitial owned by 'Send Feedback_ActivityGraphWithParam', in package 'Use Case Model'

OUTGOING BEHAVIORAL RELATIONSHIPS

⇐ ControlFlow from Start to User click "contact"

Send Feedback

Interaction owned by 'Send Feedback', in package 'Use Case Model'

Send Feedback
Version 1.0 Phase 1.0 Proposed
Lenovo created on 29-Jan-23. Last modified 29-Jan-23

ELEMENTS OWNED BY Send Feedback

ELEMENTS OWNED BY Send Feedback	
>User : Actor	
ContactPage : Sequence «boundary»	
MainController : Sequence «control»	
PlanDatabase : Sequence «entity»	

Send Feedback diagram

Interaction diagram in package 'Use Case Model'

Send Feedback
Version 1.0
Lenovo created on 29-Jan-23. Last modified 29-Jan-23

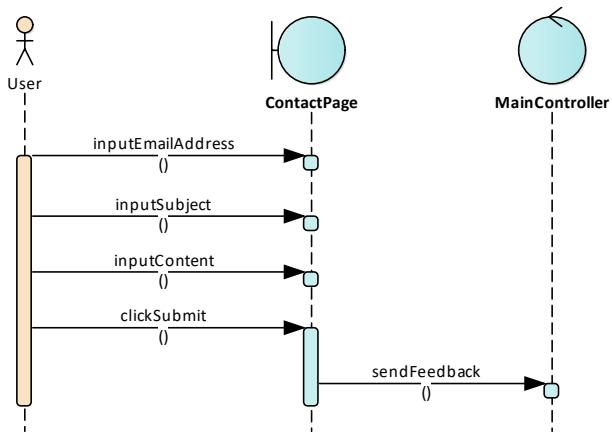


Figure 26: Send Feedback

INTERACTION MESSAGES	
✉ 1.0 'inputEmailAddress' from 'User' sent to 'ContactPage'.	Synchronous Call. Returns void. [Return is False. Iteration is False. New group is False.]
✉ 1.1 'inputSubject' from 'User' sent to 'ContactPage'.	Synchronous Call. Returns void. [Return is False. Iteration is False. New group is False.]
✉ 1.2 'inputContent' from 'User' sent to 'ContactPage'.	Synchronous Call. Returns void. [Return is False. Iteration is False. New group is False.]

1.3 'clickSubmit' from 'User' sent to 'ContactPage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

1.4 'sendFeedback' from 'ContactPage' sent to 'MainController'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

User

Actor owned by 'Send Feedback', in package 'Use Case Model'

User
Version 1.0 Phase 1.0 Proposed
Lenovo created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: inputEmailAddress

Sequence from User to «boundary» ContactPage

Name: inputSubject

Sequence from User to «boundary» ContactPage

Name: clickSubmit

Sequence from User to «boundary» ContactPage

Name: inputContent

Sequence from User to «boundary» ContactPage

ContactPage

Sequence «boundary» owned by 'Send Feedback', in package 'Use Case Model'

ContactPage
Version 1.0 Phase 1.0 Proposed
Lenovo created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: sendFeedback

Sequence from «boundary» ContactPage to «control» MainController

INCOMING BEHAVIORAL RELATIONSHIPS

Name: inputEmailAddress

Sequence from User to «boundary» ContactPage

INCOMING BEHAVIORAL RELATIONSHIPS
Name: inputSubject  Sequence from User to «boundary» ContactPage
Name: clickSubmit  Sequence from User to «boundary» ContactPage
Name: inputContent  Sequence from User to «boundary» ContactPage

MainController

Sequence «control» owned by 'Send Feedback', in package 'Use Case Model'

MainController
Version 1.0 Phase 1.0 Proposed
Lenovo created on 29-Jan-23. Last modified 29-Jan-23

INCOMING BEHAVIORAL RELATIONSHIPS
Name: sendFeedback  Sequence from «boundary» ContactPage to «control» MainController

PlanDatabase

Sequence «entity» owned by 'Send Feedback', in package 'Use Case Model'

PlanDatabase
Version 1.0 Phase 1.0 Proposed
Lenovo created on 29-Jan-23. Last modified 29-Jan-23

View Optimized Plan

UseCase in package 'Use Case Model'

View Optimized Plan
Version 1.0 Phase 1.0 Proposed
ASUS created on 29-Jan-23. Last modified 29-Jan-23

ELEMENTS OWNED BY View Optimized Plan
 View Optimized Plan_ActivityGraph : Activity
 View Optimized Plan_SequenceDiagram : Interaction
SCENARIOS

SCENARIOS

Basic Path. Basic Path

1. User selects the View Optimized Details button of a record
2. System requests the record data from database
3. System uses the PuLP linear programming solver to create an optimized results
4. System displays the plan details page

CONSTRAINTS

Pre-condition. The user is logged in to the system

[Mandatory, Weight is 0.]

Post-condition. The system successfully displays the selected optimized plan

[Mandatory, Weight is 1.]

CONNECTORS

Extend «extend» Source -> Destination

From: View Optimized Plan : UseCase, Public
To: Input Hiring Cost : UseCase, Public

Extend «extend» Source -> Destination

From: View Optimized Plan : UseCase, Public
To: Input Holding Cost per Unit : UseCase, Public

Extend «extend» Source -> Destination

From: View Optimized Plan : UseCase, Public
To: Input Permanent Worker Production : UseCase, Public

Extend «extend» Source -> Destination

From: View Optimized Plan : UseCase, Public
To: Input Temporary Worker Production : UseCase, Public

Extend «extend» Source -> Destination

From: View Optimized Plan : UseCase, Public
To: Input Firing Cost : UseCase, Public

Extend «extend» Source -> Destination

From: View Optimized Plan : UseCase, Public
To: Input Demand : UseCase, Public

Extend «extend» Source -> Destination

From: View Optimized Plan : UseCase, Public
To: Input Number of Permanent Worker : UseCase, Public

Extend «extend» Source -> Destination

From: View Optimized Plan : UseCase, Public
To: Input Starting and Ending Inventory Level : UseCase, Public

CONNECTORS **UseCaseLink** Source -> Destination

From: User : Actor, Public

To: View Optimized Plan : UseCase, Public

View Optimized Plan_ActivityGraph*Activity owned by 'View Optimized Plan', in package 'Use Case Model'*

View Optimized Plan_ActivityGraph

Version 1.0 Phase 1.0 Proposed

KingZexx created on 29-Jan-23. Last modified 29-Jan-23

ELEMENTS OWNED BY View Optimized Plan_ActivityGraph

-  System displays the plan details page : Activity
-  System requests the record data from database : Activity
-  System uses the PuLP linear programming solver to create an optimized results : Activity
-  User selects the View Optimized Details button of a record : Activity
-  End : ActivityFinal
-  Start : ActivityInitial

View Optimized Plan_ActivityGraph diagram*Activity diagram in package 'Use Case Model'*

View Optimized Plan_ActivityGraph

Version 1.0

KingZexx created on 29-Jan-23. Last modified 29-Jan-23

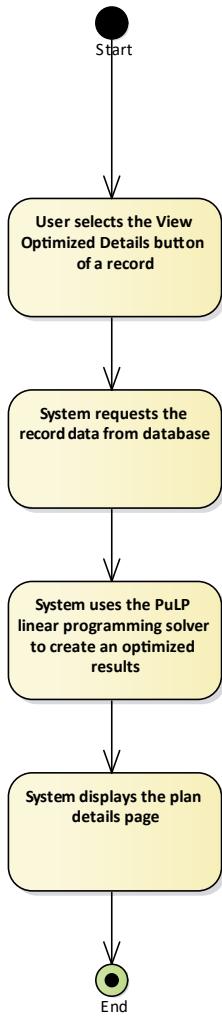


Figure 27: View Optimized Plan_ActivityGraph

System displays the plan details page

Activity owned by 'View Optimized Plan_ActivityGraph', in package 'Use Case Model'

System displays the plan details page
 Version 1.0 Phase 1.0 Proposed
 KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from System displays the plan details page to End

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from System uses the PuLP linear programming solver to create an optimized results to System displays the plan details page

System requests the record data from database

Activity owned by 'View Optimized Plan_ActivityGraph', in package 'Use Case Model'

System requests the record data from database
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from System requests the record data from database to System uses the PuLP linear programming solver to create an optimized results

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from User selects the View Optimized Details button of a record to System requests the record data from database

System uses the PuLP linear programming solver to create an optimized results

Activity owned by 'View Optimized Plan_ActivityGraph', in package 'Use Case Model'

System uses the PuLP linear programming solver to create an optimized results
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from System uses the PuLP linear programming solver to create an optimized results to System displays the plan details page

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from System requests the record data from database to System uses the PuLP linear programming solver to create an optimized results

User selects the View Optimized Details button of a record

Activity owned by 'View Optimized Plan_ActivityGraph', in package 'Use Case Model'

User selects the View Optimized Details button of a record
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

↳ ControlFlow from User selects the View Optimized Details button of a record to System requests the record data from database

INCOMING BEHAVIORAL RELATIONSHIPS

⇒ ControlFlow from Start to User selects the View Optimized Details button of a record

End

ActivityFinal owned by 'View Optimized Plan_ActivityGraph', in package 'Use Case Model'

INCOMING BEHAVIORAL RELATIONSHIPS

- ➡ ControlFlow from System displays the plan details page to End

Start

ActivityInitial owned by 'View Optimized Plan_ActivityGraph', in package 'Use Case Model'

OUTGOING BEHAVIORAL RELATIONSHIPS

- ↳ ControlFlow from Start to User selects the View Optimized Details button of a record

View Optimized Plan_SequenceDiagram

Interaction owned by 'View Optimized Plan', in package 'Use Case Model'

View Optimized Plan_SequenceDiagram
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

ELEMENTS OWNED BY View Optimized Plan_SequenceDiagram

- ▣ Variable Inputs : InteractionFragment
- ▣ 1_Basic_Path : InteractionOccurrence
- ▣ Input Demand : InteractionOccurrence
- ▣ Input Firing Cost : InteractionOccurrence
- ▣ Input Hiring Cost : InteractionOccurrence
- ▣ Input Holding Cost per Unit : InteractionOccurrence
- ▣ Input Number of Permanent Worker : InteractionOccurrence
- ▣ Input Number of Permanent Worker : InteractionOccurrence

ELEMENTS OWNED BY View Optimized Plan_SequenceDiagram	
■	Input Permanent Worker Production : InteractionOccurrence
■	Input Starting and Ending Inventory Level : InteractionOccurrence
■	Input Temporary Worker Production : InteractionOccurrence
■	Input Temporary Worker Production : InteractionOccurrence
■	HistoryPage : Sequence «boundary»
■	MainController : Sequence «control»
■	PlanDatabase : Sequence «entity»
■	PlanDetailsPage : Sequence «boundary»
■	User : Sequence

1_Basic_Path diagram

Interaction diagram in package 'Use Case Model'

1_Basic_Path
Version 1.0

KingZexx created on 29-Jan-23. Last modified 30-Jan-23

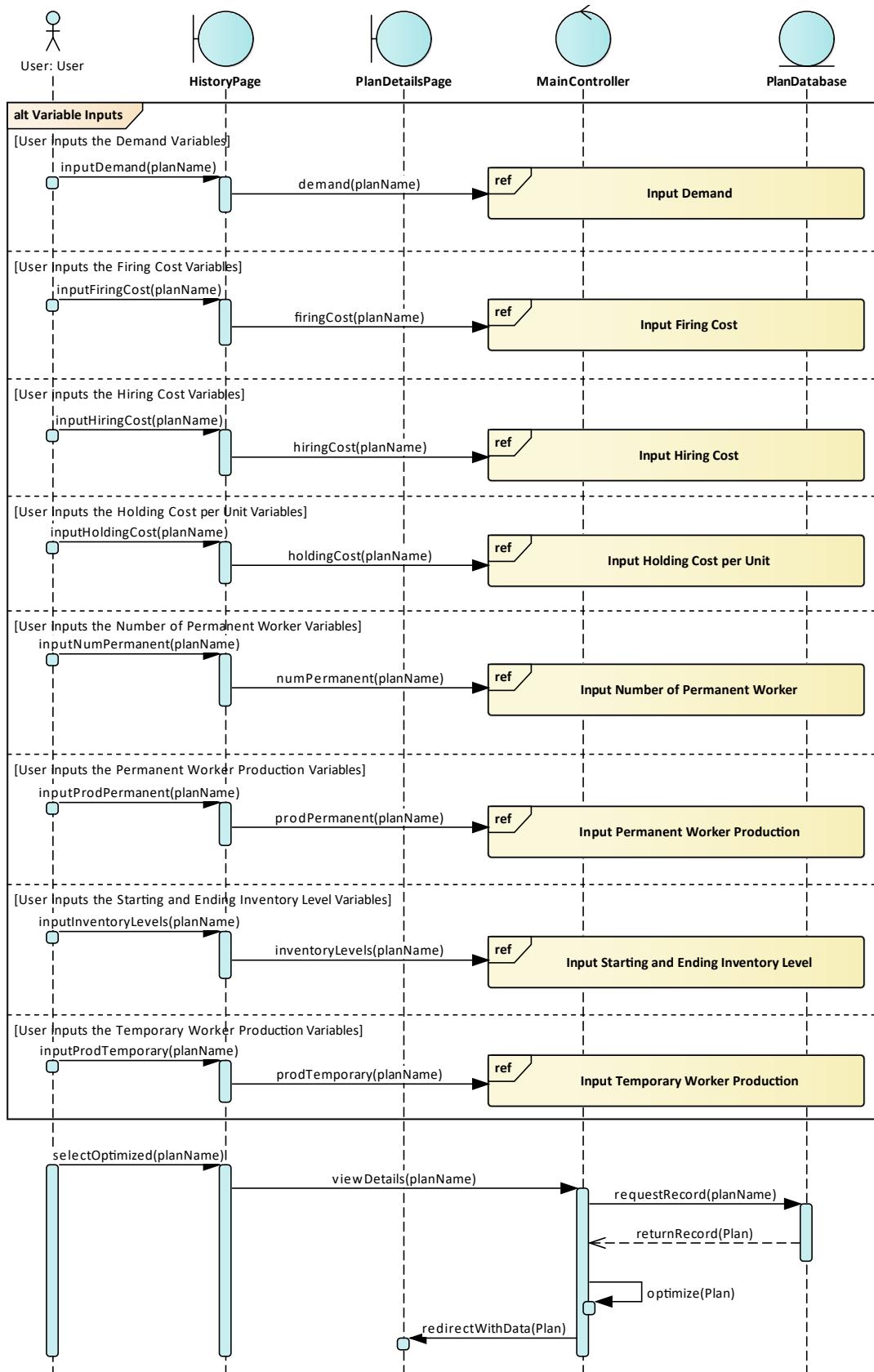


Figure 28: 1_Basic_Path

INTERACTION MESSAGES

✉ 1.0 'inputDemand' from 'User: User' sent to 'HistoryPage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.1 'demand' from 'HistoryPage' sent to 'Input Demand: Input Demand_SequenceDiagram'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.2 'inputFiringCost' from 'User: User' sent to 'HistoryPage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.3 'firingCost' from 'HistoryPage' sent to 'Input Firing Cost: Input Firing Cost_SequenceDiagram'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.4 'inputHiringCost' from 'User: User' sent to 'HistoryPage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.5 'hiringCost' from 'HistoryPage' sent to 'Input Hiring Cost: Input Hiring Cost_SequenceDiagram'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.6 'inputHoldingCost' from 'User: User' sent to 'HistoryPage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.7 'holdingCost' from 'HistoryPage' sent to 'Input Holding Cost per Unit: Input Holding Cost per Unit_SequenceDiagram'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.8 'inputNumPermanent' from 'User: User' sent to 'HistoryPage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.9 'numPermanent' from 'HistoryPage' sent to 'Input Number of Permanent Worker: Input Number of Permanent Worker_SequenceDiagram'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.10 'inputProdPermanent' from 'User: User' sent to 'HistoryPage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.11 'prodPermanent' from 'HistoryPage' sent to 'Input Permanent Worker Production: Input Permanent Worker Production_SequenceDiagram'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.12 'inputInventoryLevels' from 'User: User' sent to 'HistoryPage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.13 'inventoryLevels' from 'HistoryPage' sent to 'Input Starting and Ending Inventory Level: Input Starting and Ending Inventory Level_SequenceDiagram'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.14 'inputProdTemporary' from 'User: User' sent to 'HistoryPage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.15 'prodTemporary' from 'HistoryPage' sent to 'Input Temporary Worker Production: Input Temporary Worker Production_SequenceDiagram'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.16 'selectOptimized' from 'User: User' sent to 'HistoryPage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.17 'viewDetails' from 'HistoryPage' sent to 'MainController'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.18 'requestRecord' from 'MainController' sent to 'PlanDatabase'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.19 'returnRecord' from 'PlanDatabase' sent to 'MainController'.

Synchronous Call. Returns void.

[Return is True. Iteration is False. New group is False.]

✉ 1.20 'optimize' from 'MainController' sent to 'MainController'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

✉ 1.21 'redirectToData' from 'MainController' sent to 'PlanDetailsPage'.

Synchronous Call. Returns void.

[Return is False. Iteration is False. New group is False.]

Variable Inputs

InteractionFragment owned by 'View Optimized Plan_SequenceDiagram', in package 'Use Case Model'

Variable Inputs
Version 1.0 Phase 1.0 Proposed
ASUS created on 30-Jan-23. Last modified 30-Jan-23

1_Basic_Path

InteractionOccurrence owned by 'View Optimized Plan_SequenceDiagram', in package 'Use Case Model'

1_Basic_Path
Version 1.0 Phase 1.0 Proposed
ASUS created on 30-Jan-23. Last modified 30-Jan-23

Input Demand

InteractionOccurrence owned by 'View Optimized Plan_SequenceDiagram', in package 'Use Case Model'

Input Demand
Version 1.0 Phase 1.0 Proposed
ASUS created on 30-Jan-23. Last modified 30-Jan-23

INCOMING BEHAVIORAL RELATIONSHIPS

Name: demand

⇒ Sequence from «boundary» HistoryPage to Input Demand

Input Firing Cost

InteractionOccurrence owned by 'View Optimized Plan_SequenceDiagram', in package 'Use Case Model'

Input Firing Cost
Version 1.0 Phase 1.0 Proposed
ASUS created on 30-Jan-23. Last modified 30-Jan-23

INCOMING BEHAVIORAL RELATIONSHIPS

Name: firingCost

⇒ Sequence from «boundary» HistoryPage to Input Firing Cost

INCOMING BEHAVIORAL RELATIONSHIPS**Input Hiring Cost***InteractionOccurrence owned by 'View Optimized Plan_SequenceDiagram', in package 'Use Case Model'*

Input Hiring Cost
Version 1.0 Phase 1.0 Proposed
ASUS created on 30-Jan-23. Last modified 30-Jan-23

INCOMING BEHAVIORAL RELATIONSHIPS

Name: hiringCost
⇒ Sequence from «boundary» HistoryPage to Input Hiring Cost

Input Holding Cost per Unit*InteractionOccurrence owned by 'View Optimized Plan_SequenceDiagram', in package 'Use Case Model'*

Input Holding Cost per Unit
Version 1.0 Phase 1.0 Proposed
ASUS created on 30-Jan-23. Last modified 30-Jan-23

INCOMING BEHAVIORAL RELATIONSHIPS

Name: holdingCost
⇒ Sequence from «boundary» HistoryPage to Input Holding Cost per Unit

Input Number of Permanent Worker*InteractionOccurrence owned by 'View Optimized Plan_SequenceDiagram', in package 'Use Case Model'*

Input Number of Permanent Worker
Version 1.0 Phase 1.0 Proposed
ASUS created on 30-Jan-23. Last modified 30-Jan-23

Input Number of Permanent Worker*InteractionOccurrence owned by 'View Optimized Plan_SequenceDiagram', in package 'Use Case Model'*

Input Number of Permanent Worker
Version 1.0 Phase 1.0 Proposed
ASUS created on 30-Jan-23. Last modified 30-Jan-23

INCOMING BEHAVIORAL RELATIONSHIPS

Name: numPermanent
⇒ Sequence from «boundary» HistoryPage to Input Number of Permanent Worker

Input Permanent Worker Production

InteractionOccurrence owned by 'View Optimized Plan_SequenceDiagram', in package 'Use Case Model'

Input Permanent Worker Production
Version 1.0 Phase 1.0 Proposed
ASUS created on 30-Jan-23. Last modified 30-Jan-23

INCOMING BEHAVIORAL RELATIONSHIPS

Name: prodPermanent

⇒ Sequence from «boundary» HistoryPage to Input Permanent Worker Production

Input Starting and Ending Inventory Level

InteractionOccurrence owned by 'View Optimized Plan_SequenceDiagram', in package 'Use Case Model'

Input Starting and Ending Inventory Level
Version 1.0 Phase 1.0 Proposed
ASUS created on 30-Jan-23. Last modified 30-Jan-23

INCOMING BEHAVIORAL RELATIONSHIPS

Name: inventoryLevels

⇒ Sequence from «boundary» HistoryPage to Input Starting and Ending Inventory Level

Input Temporary Worker Production

InteractionOccurrence owned by 'View Optimized Plan_SequenceDiagram', in package 'Use Case Model'

Input Temporary Worker Production
Version 1.0 Phase 1.0 Proposed
ASUS created on 30-Jan-23. Last modified 30-Jan-23

INCOMING BEHAVIORAL RELATIONSHIPS

Name: prodTemporary

⇒ Sequence from «boundary» HistoryPage to Input Temporary Worker Production

Input Temporary Worker Production

InteractionOccurrence owned by 'View Optimized Plan_SequenceDiagram', in package 'Use Case Model'

Input Temporary Worker Production
Version 1.0 Phase 1.0 Proposed
ASUS created on 30-Jan-23. Last modified 30-Jan-23

HistoryPage

Sequence «boundary» owned by 'View Optimized Plan_SequenceDiagram', in package 'Use Case Model'

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: hiringCost

↳ Sequence from «boundary» HistoryPage to Input Hiring Cost

Name: holdingCost

↳ Sequence from «boundary» HistoryPage to Input Holding Cost per Unit

Name: prodPermanent

↳ Sequence from «boundary» HistoryPage to Input Permanent Worker Production

Name: viewDetails

↳ Sequence from «boundary» HistoryPage to «control» MainController

Name: numPermanent

↳ Sequence from «boundary» HistoryPage to Input Number of Permanent Worker

Name: prodTemporary

↳ Sequence from «boundary» HistoryPage to Input Temporary Worker Production

Name: inventoryLevels

↳ Sequence from «boundary» HistoryPage to Input Starting and Ending Inventory Level

Name: firingCost

↳ Sequence from «boundary» HistoryPage to Input Firing Cost

Name: demand

↳ Sequence from «boundary» HistoryPage to Input Demand

INCOMING BEHAVIORAL RELATIONSHIPS

Name: inputFiringCost

⇒ Sequence from User to «boundary» HistoryPage

Name: inputNumPermanent

⇒ Sequence from User to «boundary» HistoryPage

Name: inputInventoryLevels

⇒ Sequence from User to «boundary» HistoryPage

Name: inputDemand

⇒ Sequence from User to «boundary» HistoryPage

INCOMING BEHAVIORAL RELATIONSHIPS	
Name: inputHoldingCost	 Sequence from User to «boundary» HistoryPage
Name: inputProdTemporary	 Sequence from User to «boundary» HistoryPage
Name: inputProdPermanent	 Sequence from User to «boundary» HistoryPage
Name: inputHiringCost	 Sequence from User to «boundary» HistoryPage
Name: selectOptimized	 Sequence from User to «boundary» HistoryPage

MainController

Sequence «control» owned by 'View Optimized Plan_SequenceDiagram', in package 'Use Case Model'

MainController
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS	
Name: redirectWithData	 Sequence from «control» MainController to «boundary» PlanDetailsPage
Name: optimize	 Sequence from «control» MainController to «control» MainController
Name: requestRecord	 Sequence from «control» MainController to «entity» PlanDatabase

INCOMING BEHAVIORAL RELATIONSHIPS	
Name: optimize	 Sequence from «control» MainController to «control» MainController
Name: viewDetails	 Sequence from «boundary» HistoryPage to «control» MainController
Name: returnRecord	 Sequence from «entity» PlanDatabase to «control» MainController

PlanDatabase

Sequence «entity» owned by 'View Optimized Plan_SequenceDiagram', in package 'Use Case Model'

PlanDatabase
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: returnRecord
↳ Sequence from «entity» PlanDatabase to «control» MainController

INCOMING BEHAVIORAL RELATIONSHIPS

Name: requestRecord
↗ Sequence from «control» MainController to «entity» PlanDatabase

PlanDetailsPage

Sequence «boundary» owned by 'View Optimized Plan_SequenceDiagram', in package 'Use Case Model'

PlanDetailsPage
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

INCOMING BEHAVIORAL RELATIONSHIPS

Name: redirectWithData
↗ Sequence from «control» MainController to «boundary» PlanDetailsPage

User

Sequence owned by 'View Optimized Plan_SequenceDiagram', in package 'Use Case Model'

User
Version 1.0 Phase 1.0 Proposed
KingZexx created on 29-Jan-23. Last modified 29-Jan-23

OUTGOING BEHAVIORAL RELATIONSHIPS

Name: inputFiringCost
↳ Sequence from User to «boundary» HistoryPage

Name: inputNumPermanent
↳ Sequence from User to «boundary» HistoryPage

Name: inputInventoryLevels
↳ Sequence from User to «boundary» HistoryPage

OUTGOING BEHAVIORAL RELATIONSHIPS	
Name: inputDemand	↳ Sequence from User to «boundary» HistoryPage
Name: inputHoldingCost	↳ Sequence from User to «boundary» HistoryPage
Name: inputProdTemporary	↳ Sequence from User to «boundary» HistoryPage
Name: inputProdPermanent	↳ Sequence from User to «boundary» HistoryPage
Name: inputHiringCost	↳ Sequence from User to «boundary» HistoryPage
Name: selectOptimized	↳ Sequence from User to «boundary» HistoryPage