Supporting Code Comprehension via Annotations: Right Information at the Right Time and Place

Marjan Adeli, **Nicholas Nelson**, Souti Chattopadhyay, Hayden Coffey, Austin Henley, and Anita Sarma

{adelima, **nelsonni**, chattops, anita.sarma}@oregonstate.edu

khcoffey1@vols.utk.edu, azh@utk.edu

Oregon State University

THE UNIVERSITY OF TENNESSEE KNOXVILLE

@nomaticdev

https://nomatic.dev/

Code Comprehension

- Code Comprehension:
  - Navigating through a codebase
  - Building a mental model of that code
- Large portion of developers' activities
  - 58% of the effort [1]

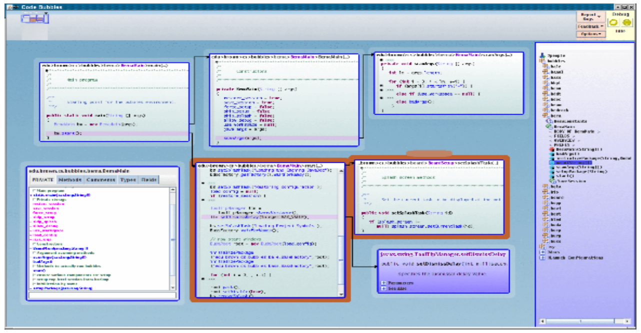
[1] Xia, X., Bao, L., Lo, D., Xing, Z., Hassan, A. E., & Li, S. (2017). Measuring program comprehension: A large-scale field study with professionals. *IEEE Transactions on Software Engineering*, *44*(10), 951-976.

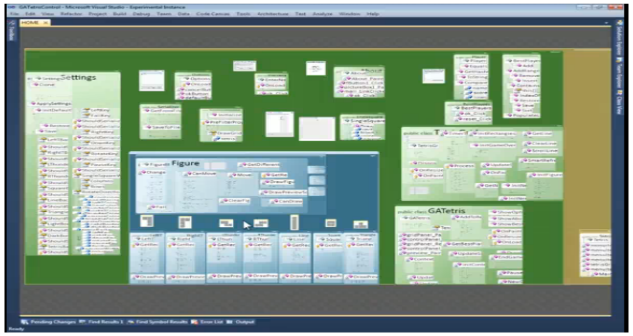
Difficulties in code comprehension

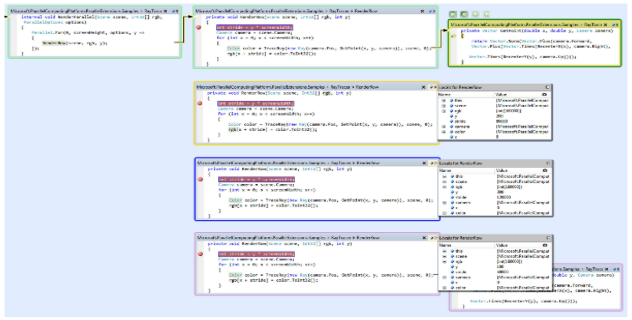
Understanding code requires developers to:

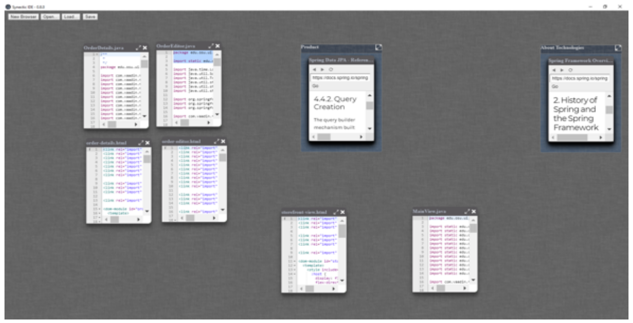
- Manage different types of artifacts
- Locate relevant information in different places
- Understand the relationships between artifacts

Facilitating code comprehension in IDEs


Code Bubbles

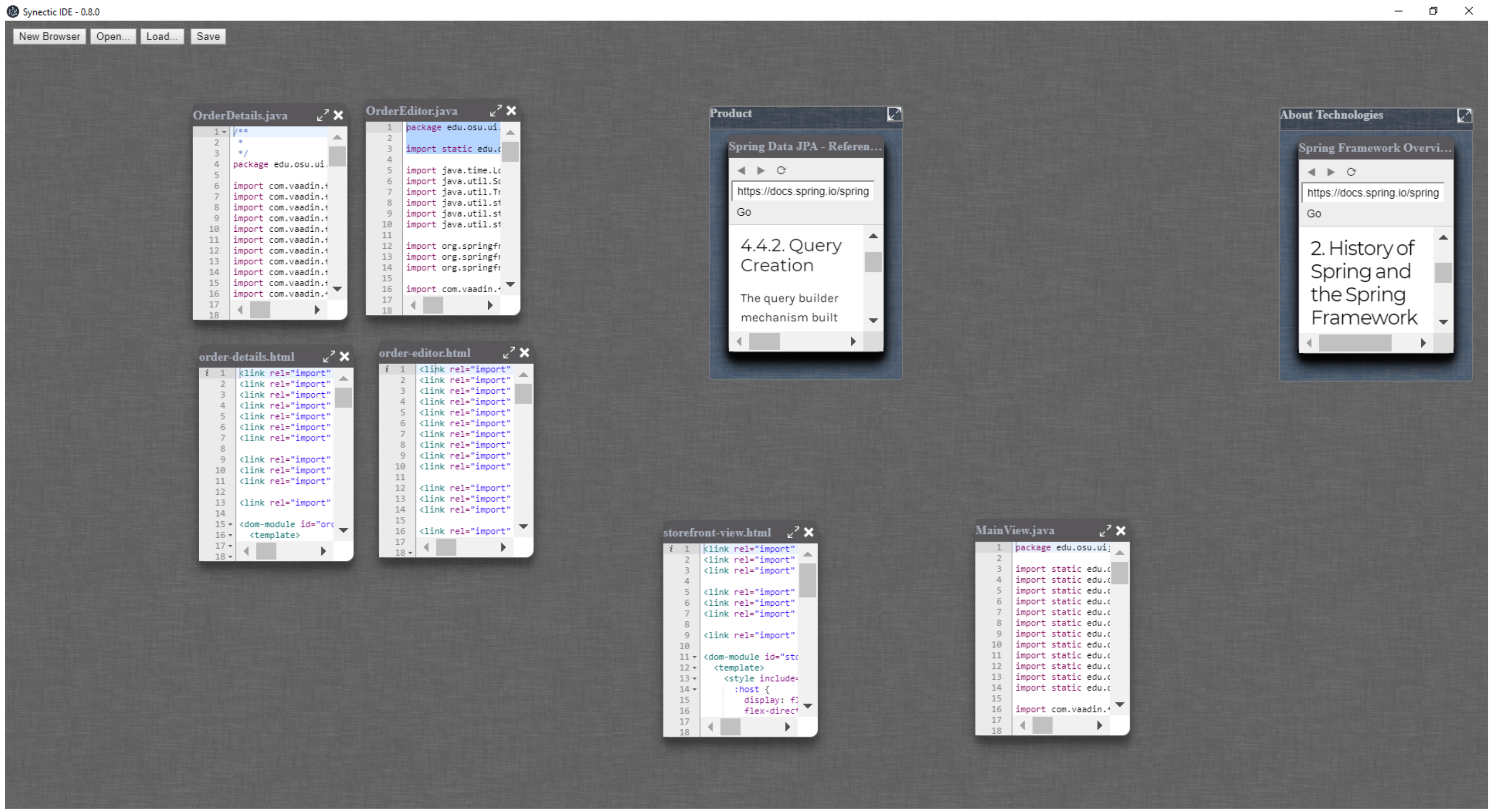

Code Canvas


Debugger Canvas


Synectic

●●●●○○○○○○○○○○○○○

Synectic : A canvas-based IDE



OrderDetails.java

```
/**
 *
 */
package edu.osu.ui.

import com.vaadin.
import com.vaadin.
import com.vaadin.
import com.vaadin.
import com.vaadin.
import com.vaadin.
import com.vaadin.
import com.vaadin.
import com.vaadin.
import com.vaadin.
import com.vaadin.
```

OrderEditor.java

```
package edu.osu.ui.

import static edu.c

import java.time.Lc
import java.util.Sc
import java.util.Tr
import java.util.st
import java.util.st
import java.util.st

import org.springfr
import org.springfr
import org.springfr

import com.vaadin.
```

order-details.html

```
<link rel="import"
<link rel="import"
<link rel="import"
<link rel="import"
<link rel="import"
<link rel="import"
<link rel="import"

<link rel="import"
<link rel="import"
<link rel="import"

<link rel="import"

<dom-module id="orc
  <template>
```

order-editor.html

```
<link rel="import"
<link rel="import"
<link rel="import"
<link rel="import"
<link rel="import"
<link rel="import"
<link rel="import"
<link rel="import"
<link rel="import"
<link rel="import"

<link rel="import"
<link rel="import"
<link rel="import"

<link rel="import"
```

https://docs.spring.io/spring

4.4.2. Query Creation

The query builder mechanism built

https://docs.spring.io/spring

2. History of Spring and the Spring Framework

storefront-view.html

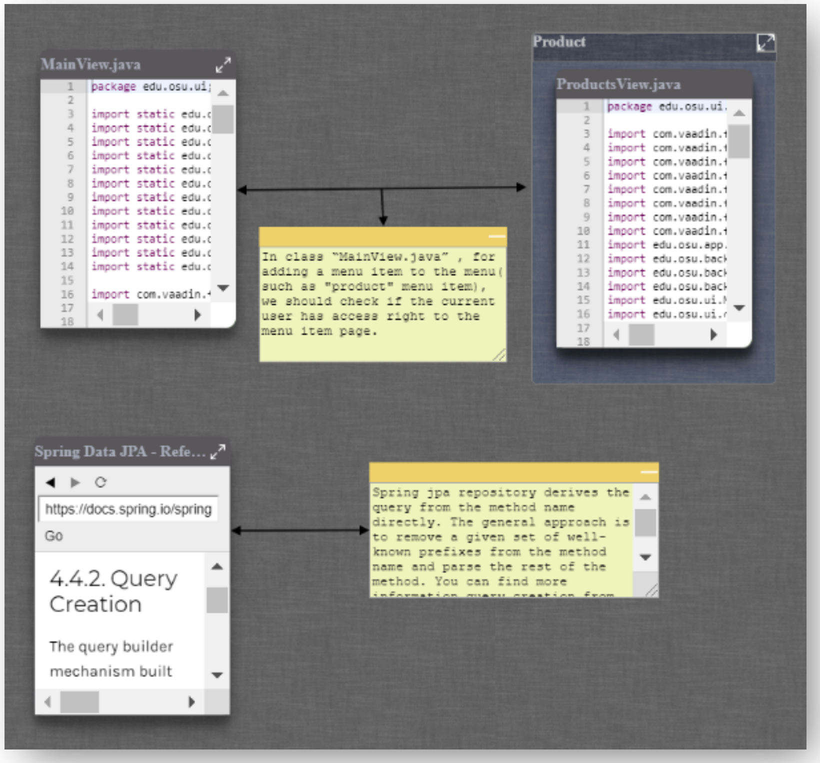
```
<link rel="import"
<link rel="import"
<link rel="import"

<link rel="import"
<link rel="import"
<link rel="import"

<link rel="import"

<dom-module id="stc
  <template>
    <style include=
      :host {
        display: fl
        flex-direct
```

MainView.java

```
package edu.osu.ui;

import static edu.c
import static edu.c
import static edu.c
import static edu.c
import static edu.c
import static edu.c
import static edu.c
import static edu.c
import static edu.c
import static edu.c
import static edu.c
import static edu.c

import com.vaadin.
```

Annotation Overlay

- **Annotation notes** for capturing design rationale, expected API usage patterns, corner-cases, etc.
- **Annotation links** for connecting notes to cards or groups
- Multiway connections between annotations and cards to describe relationships.


MainView.java
Product
ProductsView.java
Spring Data JPA - Refe...
https://docs.spring.io/spring
Go
4.4.2. Query Creation
The query builder mechanism built

In class "MainView.java" , for adding a menu item to the menu(such as "product" menu item), we should check if the current user has access right to the menu item page.

Spring jpa repository derives the query from the method name directly. The general approach is to remove a given set of well-known prefixes from the method name and parse the rest of the method. You can find more

Study Design - RQs

- RQ1: How do annotations affect code comprehension among newcomers?
  - Do annotations increase the accuracy of responses?
  - Do annotations reduce the time to task completion?
  - Do annotations reduce cognitive load?

User Study

Controlled lab study

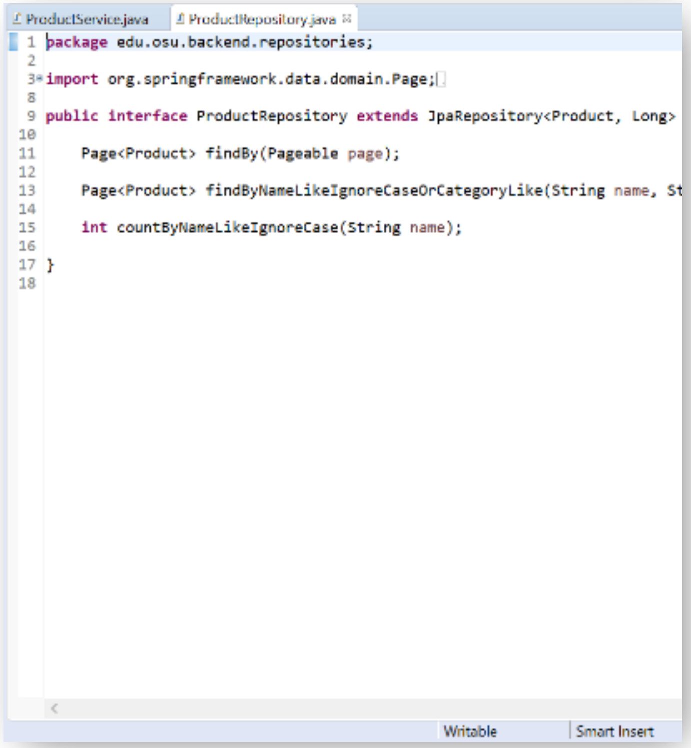
- Between-subject design
- 22 participants (graduate students)
- 4 code comprehension tasks

Synectic treatment	**Eclipse treatment**
11 participants	11 participants
4 code comprehension tasks	4 code comprehension tasks
Onboarding information added as annotations	Onboarding information added as text document

Study Design – Tasks

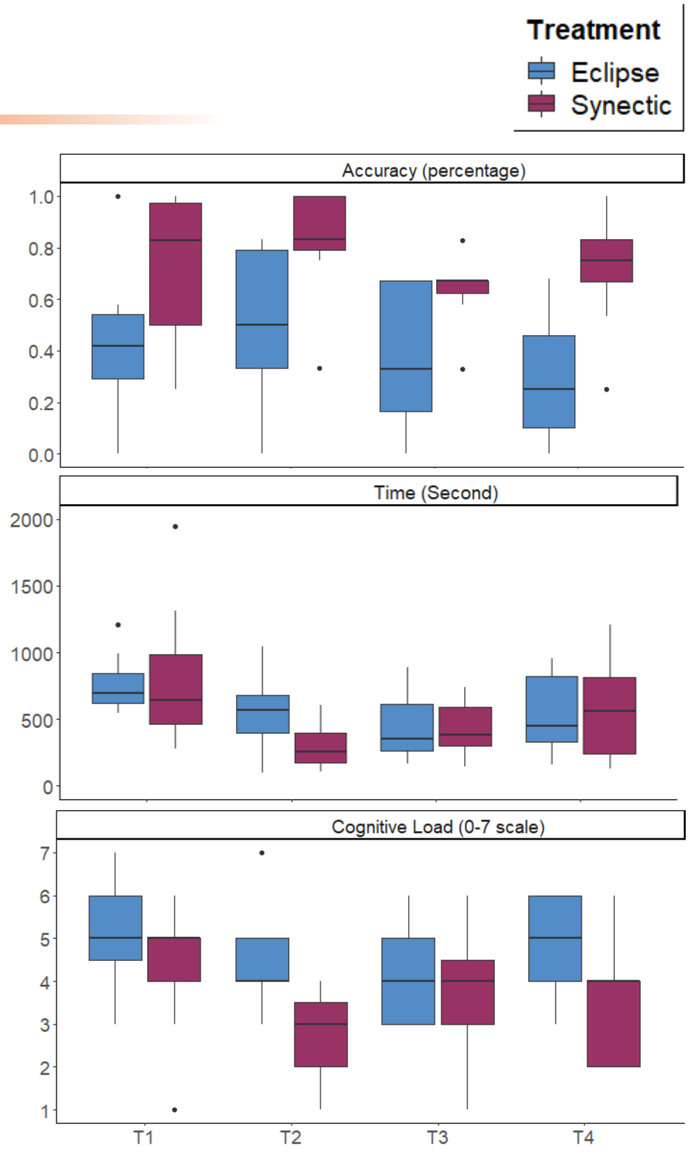
- Code Comprehension Task
  - Navigation portion
  - Comprehension portion

- Designed as onboarding tasks
  - Locating code related to a feature
  - Learning how to make changes to those features



```java
package edu.osu.backend.repositories;

import org.springframework.data.domain.Page;

public interface ProductRepository extends JpaRepository<Product, Long>

    Page<Product> findBy(Pageable page);

    Page<Product> findByNameLikeIgnoreCaseOrCategoryLike(String name, St

    int countByNameLikeIgnoreCase(String name);

}

```

Results



Accuracy

Rank Based Non-Parametric (RBNP) ANOVA test
(p-value < 0.001, statistic = 19.46488)

Time

RBNP ANOVA test
(p-value = 0.22, statistic = 1.607723)

Cognitive Load

RBNP ANOVA test
(p-value = 0.003, statistic = 11.52591)

Discussion

- Quantitative results
  - Accuracy & Cognitive Load differences were significant
  - Time differences were not statistically significant

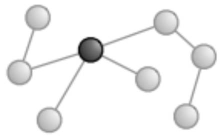
- Qualitative results
  - Sillito et al.'s four stages of comprehension model[1] to explain comprehension
  - Information Foraging Theory (IFT) to explain navigation

[1] Sillito, J., Murphy, G. C., & De Volder, K. (2008). Asking and answering questions during a programming change task. *IEEE Transactions on Software Engineering*, *34*(4), 434-451.

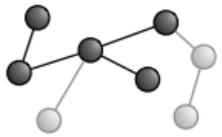
Discussion – Stages of comprehension

Sillito et al. identified 4 categories of comprehension:

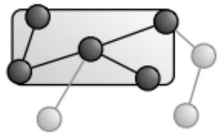
1. Finding the initial focus point



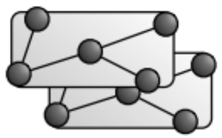
2. Building on those focus points



3. Understanding the concepts between related entities

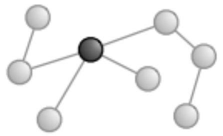


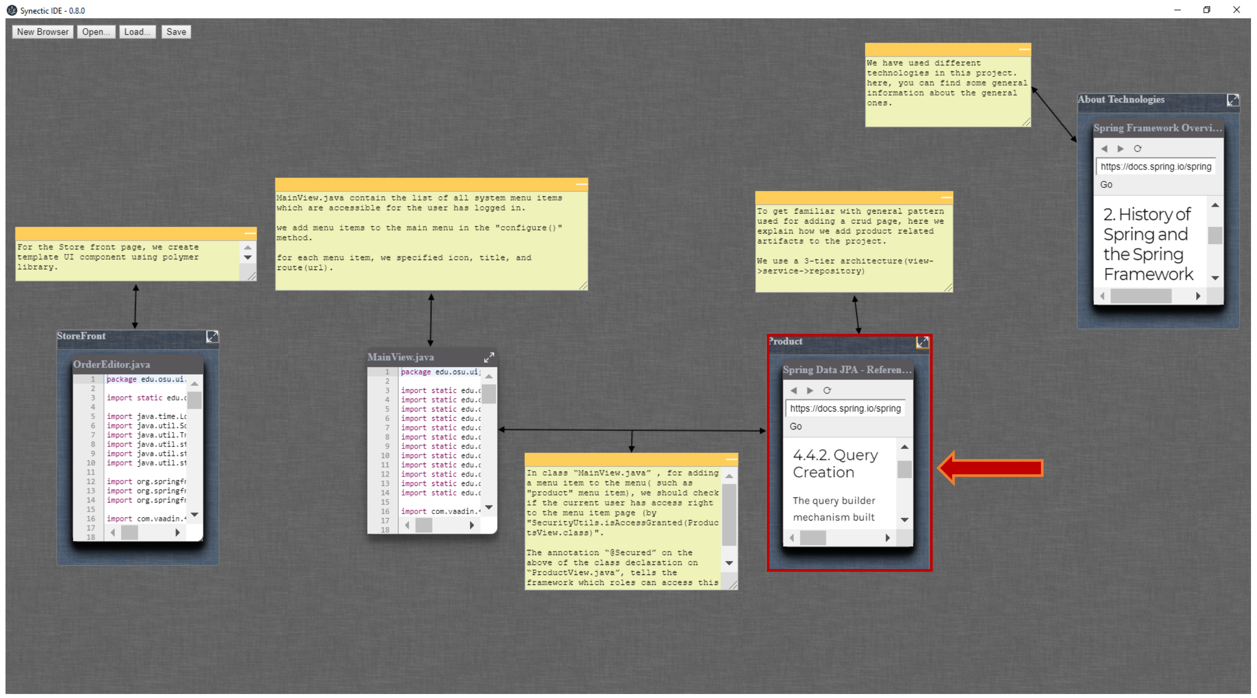
4. Understanding concepts across multiple groups of related entities



Discussion – Stages of comprehension

1. Finding the initial focus point





Synectic IDE - 0.8.0
New Browser
Open...
Load...
Save
StoreFront
OrderEditor.java
MainView.java
Product
Spring Data JPA - Referen...
https://docs.spring.io/spring
Go
4.4.2. Query Creation
The query builder mechanism built
About Technologies
Spring Framework Overvi...
https://docs.spring.io/spring
Go
2. History of Spring and the Spring Framework

We have used different technologies in this project. here, you can find some general information about the general ones.

For the Store front page, we create template UI component using polymer library.

MainView.java contain the list of all system menu items which are accessible for the user has logged in.

we add menu items to the main menu in the "configure()" method.

for each menu item, we specified icon, title, and route(url).

To get familiar with general pattern used for adding a crud page, here we explain how we add product related artifacts to the project.

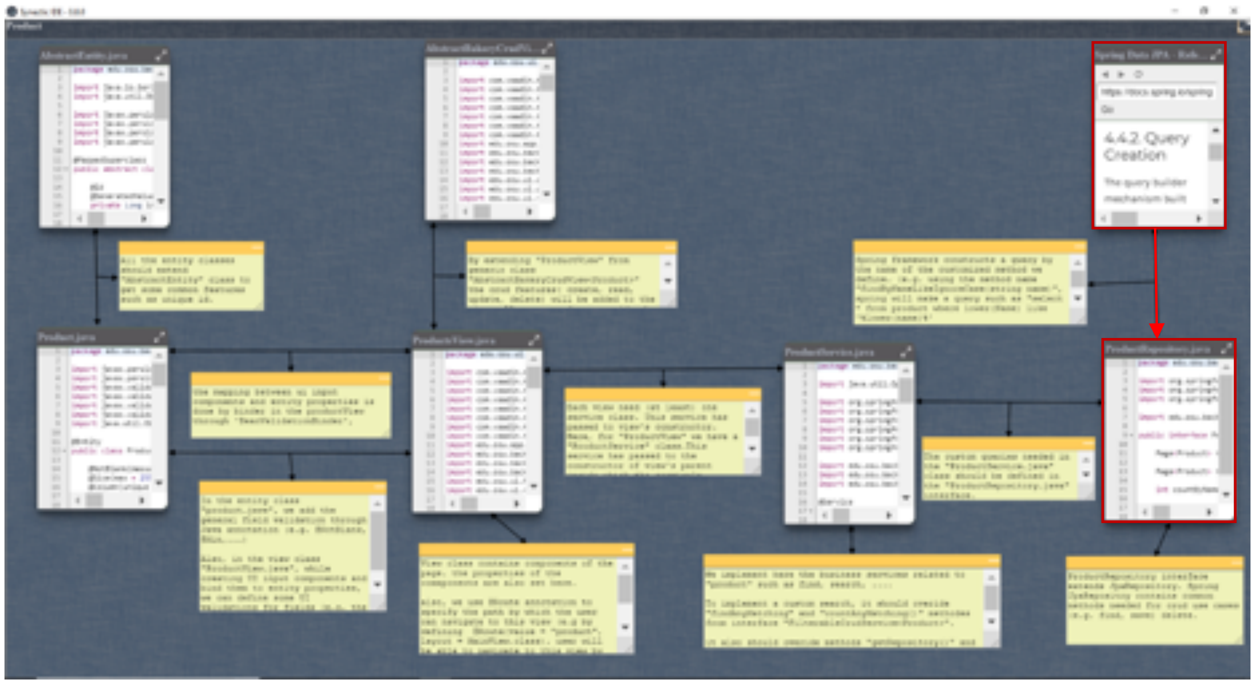
We use a 3-tier architecture(view->service->repository)

In class "MainView.java" , for adding a menu item to the menu(such as "product" menu item), we should check if the current user has access right to the menu item page (by "SecurityUtils.isAccessGranted(ProductsView.class)".

The annotation "@Secured" on the above of the class declaration on "ProductView.java", tells the framework which roles can access this

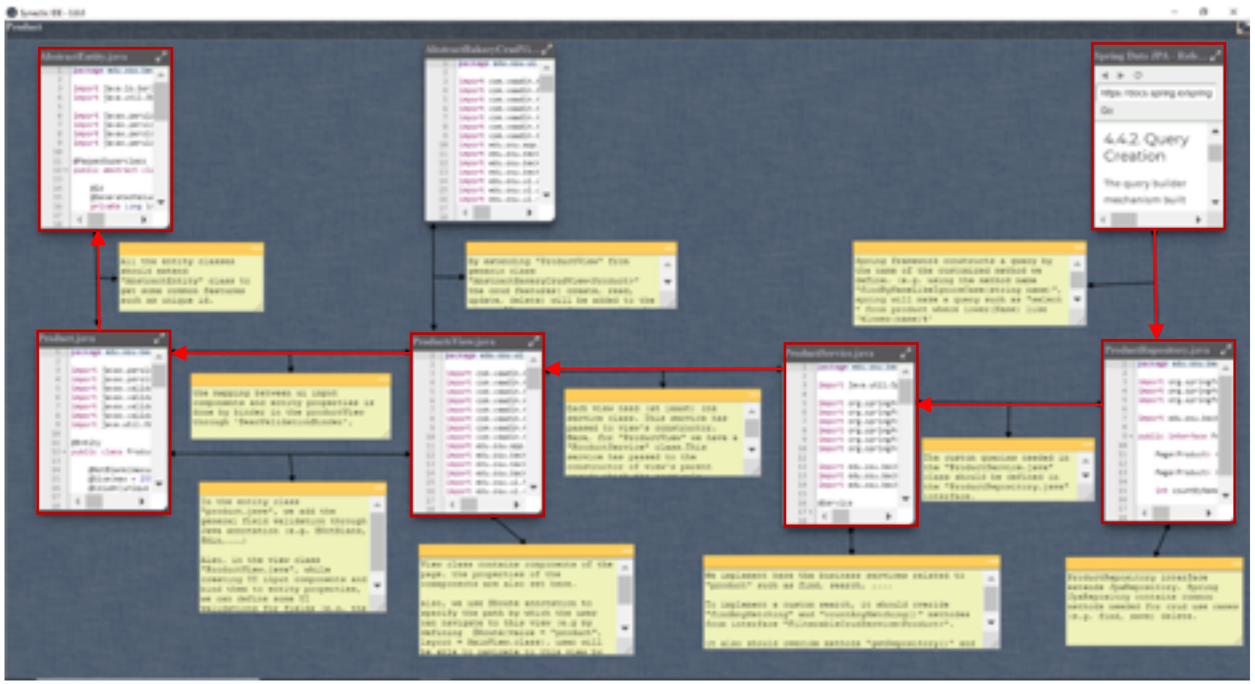
Discussion – Stages of comprehension

2. Building on those focus points



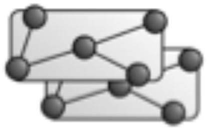
Discussion – Stages of comprehension

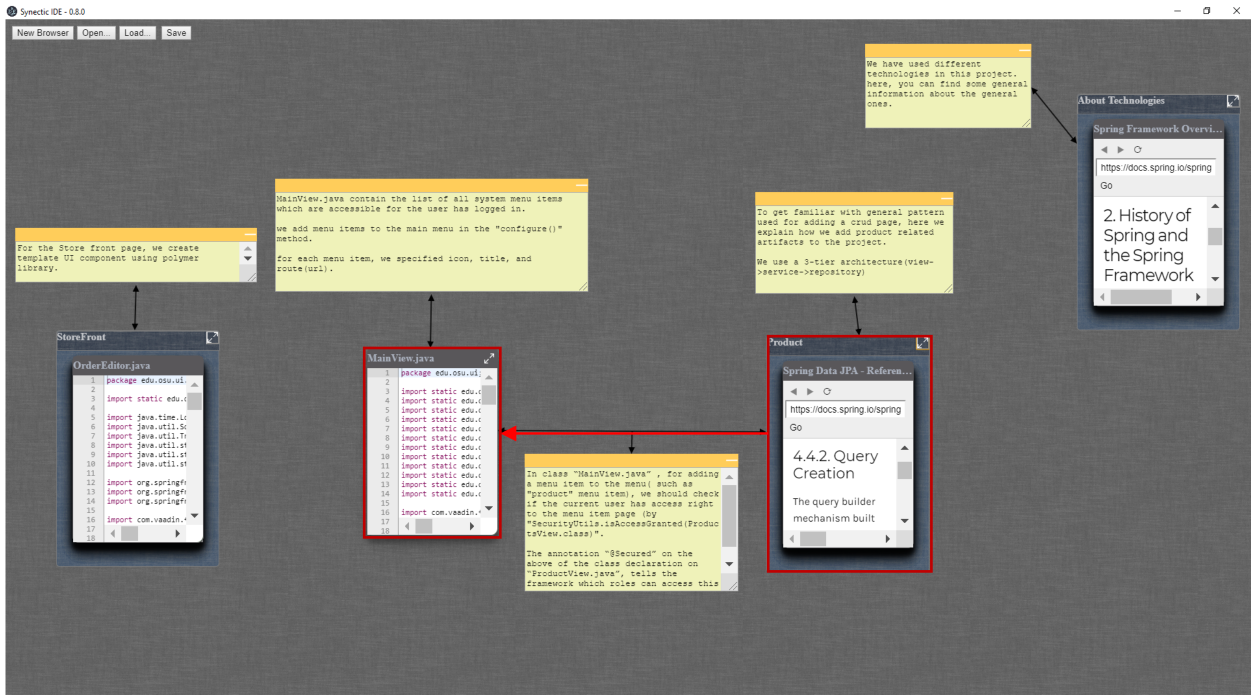
3. Understanding the concepts between related entities



Discussion – Stages of comprehension

4. Understanding concepts across multiple groups of related entities





Synectic IDE - 0.8.0
New Browser
Open...
Load...
Save
About Technologies
Spring Framework Overvi...
https://docs.spring.io/spring
Go
2. History of Spring and the Spring Framework
StoreFront
OrderEditor.java
MainView.java
Product
Spring Data JPA - Referen...
https://docs.spring.io/spring
Go
4.4.2. Query Creation
The query builder mechanism built

We have used different technologies in this project. here, you can find some general information about the general ones.

MainView.java contain the list of all system menu items which are accessible for the user has logged in.

we add menu items to the main menu in the "configure()" method.

for each menu item, we specified icon, title, and route(url).

For the Store front page, we create template UI component using polymer library.

To get familiar with general pattern used for adding a crud page, here we explain how we add product related artifacts to the project.

We use a 3-tier architecture(view->service->repository)

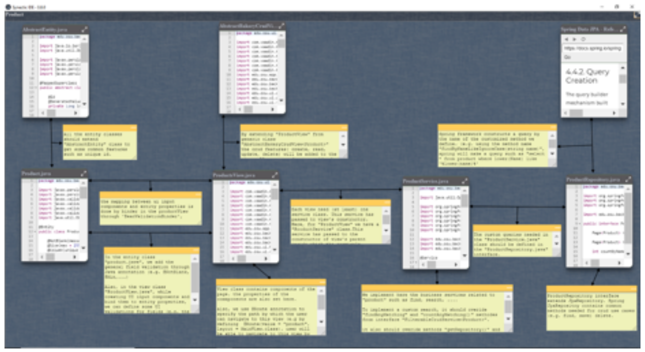
```
package edu.osu.ui.

import static edu.c

import java.time.Lc
import java.util.Sc
import java.util.Tr
import java.util.st
import java.util.st
import java.util.st

import org.springfr
import org.springfr
import org.springfr

import com.vaadin.*
```

```
package edu.osu.ui;

import static edu.c
import static edu.c
import static edu.c
import static edu.c
import static edu.c
import static edu.c
import static edu.c
import static edu.c
import static edu.c
import static edu.c
import static edu.c
import static edu.c

import com.vaadin.*
```

In class “MainView.java” , for adding a menu item to the menu(such as "product" menu item), we should check if the current user has access right to the menu item page (by "SecurityUtils.isAccessGranted(ProductsView.class)".

The annotation “@Secured” on the above of the class declaration on “ProductView.java”, tells the framework which roles can access this

Summary



- Annotations in a canvas-based IDE resulted in:
  - Lower cognitive load among newcomers
  - More accurate comprehension responses
  - Required no additional time compared traditional IDEs

- Design challenges for annotations within IDEs:
  - Manage different types of artifacts
  - Locate relevant information in different places
  - Understand the relationships between artifacts

“Right information, at the right place, and the right time”

@nomaticdev

https://nomatic.dev/