

Intro



AALBORG UNIVERSITY
DENMARK



Content

Food Planner

DS301E14

Intro

2

Problem statement &
Information Gathering

Quality Assurance

Unit Testing

Usability Testing

Reflection

Code Examples

- ▶ Causes
- ▶ Problem Statement
- ▶ XAML
- ▶ Code Exammples
- ▶ Unit Testing
- ▶ Usability Test
- ▶ Reflection

Problem statement & Information Gathering



AALBORG UNIVERSITY
DENMARK

Problem statement

Food Planner

DS301E14

Intro

Problem statement &
Information Gathering

Quality Assurance

Unit Testing

Usability Testing

Reflection

Code Examples

4

- How can a software system assist the consumer in planning and managing a meal plan to minimize food waste and simultaneously be flexible enough to support their planning, shopping and cooking habits?



Information Gathering

Food Planner

DS301E14



Intro

**Problem statement &
Information Gathering**

5

Quality Assurance

Unit Testing

Usability Testing

Reflection

Code Examples

Department of Computer
Science

Selma Lagerlöfs Vej 300
DK-9220 Aalborg East
<http://cs.aau.dk>

Quality Assurance



AALBORG UNIVERSITY
DENMARK



Quality assurance

Fulfilling the semester description

Food Planner

DS301E14

Intro

Problem statement &
Information Gathering

Quality Assurance

Unit Testing

Usability Testing

Reflection

Code Examples

7

- ▶ Quality assurance (in terms of program testing and results and usability evaluation and results), including arguments for coverage and validity (Semester description)
 - ▶ Program testing = Unit testing
 - ▶ Usability evaluation = IDA (report section 11)

Unit Testing



AALBORG UNIVERSITY
DENMARK



Unit testing

What is unit testing?

Food Planner

DS301E14

Intro

Problem statement &
Information Gathering

Quality Assurance

Unit Testing

Usability Testing

Reflection

Code Examples

- ▶ Initial stage of program testing
- ▶ Isolate smallest testable code
- ▶ Confirm if the unit behaves correctly

9



Unit testing

Possible candidates for unit testing

Food Planner

DS301E14

Intro

Problem statement &
Information Gathering

Quality Assurance

Unit Testing

Usability Testing

Reflection

Code Examples

10

- ▶ Classes e.g. User, Ingredient or Recipe
 - ▶ Testing their properties or methods
- ▶ List of candidates
 - ▶ MODELS
 - ▶ Graylist
 - ▶ Ingredient
 - ▶ InventoryIngredient*
 - ▶ InventoryListCombinedByQuantity
 - ▶ LastMeal
 - ▶ PublicQuerys
 - ▶ Recipe
 - ▶ RecipeIngredient
 - ▶ SearchResults
 - ▶ ShoppingClass
 - ▶ ShoppingListIngredient
 - ▶ User
 - ▶ VIEWMODELS
 - ▶ InventoryViewModel*
 - ▶ MealPlanViewModel
 - ▶ RecipeSearchViewModel

26



Unit testing

Code from InventoryIngredientTests.cs

Food Planner

DS301E14

Intro

Problem statement &
Information Gathering

Quality Assurance

Unit Testing

Usability Testing

Reflection

Code Examples

11

```
[TestMethod]
```

```
public void PurchaseDate_AutoSetConstructor_SetToM
```

```
//arrange
```

```
Ingredient testIngredient = new Ingredient();
```

```
DateTime expectedPurchaseDate = DateTime.Now;
```

```
//act – The property is set automatically in the co
```

```
InventoryIngredient testInventoryIngredient = new
```

```
//assert
```

```
Assert.AreEqual(expectedPurchaseDate, testInvento
```

```
}
```

Usability Testing



AALBORG UNIVERSITY
DENMARK



IDA

Food Planner

DS301E14

Intro

Problem statement &
Information Gathering

Quality Assurance

Unit Testing

Usability Testing

Reflection

Code Examples

- ▶ 3 persons
- ▶ Video camera
- ▶ Screen recorder
- ▶ Follow-up questions

13

26

IDA outcome

Food Planner

DS301E14

Intro

Problem statement &
Information Gathering

Quality Assurance

Unit Testing

Usability Testing

Reflection

Code Examples

14

- ▶ Critical problems
 - ▶ Scheduled meal page, update button
 - ▶ Deleting meal
 - ▶ Automatical or manual update
- ▶ Serious problems
 - ▶ Planning meal the wrong way
 - ▶ Top bar in recipe screen
 - ▶ Changing the number of days to shop for
- ▶ Cosmetic problems
 - ▶ Dividing of setting screen
 - ▶ Adding ingredient to lists
 - ▶ Too specific rating

26



Usability notes

Food Planner

DS301E14

► Waterfall vs Iterative

Intro

Problem statement &
Information Gathering

Quality Assurance

Unit Testing

Usability Testing

Reflection

Code Examples

15

Department of Computer
Science

Selma Lagerlöfs Vej 300
DK-9220 Aalborg East
<http://cs.aau.dk>

26

Reflection



AALBORG UNIVERSITY
DENMARK



missing features

Food Planner

DS301E14



Intro

Problem statement &
Information Gathering

Quality Assurance

Unit Testing

Usability Testing

Reflection

Code Examples

17

Department of Computer
Science

Selma Lagerlöfs Vej 300
DK-9220 Aalborg East
<http://cs.aau.dk>

26



Optimization

Food Planner

DS301E14



Intro

Problem statement &
Information Gathering

Quality Assurance

Unit Testing

Usability Testing

Reflection

Code Examples

18

26



Evaluation of design

Food Planner

DS301E14



Intro

Problem statement &
Information Gathering

Quality Assurance

Unit Testing

Usability Testing

Reflection

Code Examples

19

Department of Computer
Science

Selma Lagerlöfs Vej 300
DK-9220 Aalborg East
<http://cs.aau.dk>

26

Requirements

Food Planner

DS301E14

Intro

Problem statement &
Information Gathering

Quality Assurance

Unit Testing

Usability Testing

Reflection

Code Examples

20

Problem based requirements

- ▶ To much food being wasted
- ▶ Miscalculations happen in groups
- ▶ Shopping is time consuming
- ▶ Diets are difficult
- ▶ Consumers living alone have the most food waste
- ▶ Consumers living alone have a higher cost per meal

26

Code Examples



AALBORG UNIVERSITY
DENMARK

Sorting

Food Planner

DS301E14

Intro

Problem statement &
Information Gathering

Quality Assurance

Unit Testing

Usability Testing

Reflection

Code Examples

22

- ▶ Percentage Full Match
- ▶ Percentage Partial Match
- ▶ Rating
- ▶ Previous Ingredients
- ▶ Recipe Title

26

Food Planner

DS301E14

Intro

Problem statement & Information Gathering

Quality Assurance

Unit Testing

Usability Testing

Reflection

Code Examples

23

```

17 30 #region Fields
18 //A query that if multiple of the same ingredient is found in the users inventory, combines the sum into a single item, and combine the quantities onto one.
19 public IQueryable<InventoryListGroupedByQuantity> InventoryIQueryable =
20     from ii in App.db.InventoryIngredients
21     join i in App.db.Ingredients on ii.IngredientID equals i.ID
22     where ii.UserID == App.CurrentUser.ID
23     group ii by ii.IngredientID into iig
24     select new InventoryListGroupedByQuantity()
25     {
26         IngredientID = iig.FirstOrDefault().IngredientID,
27         Unit = iig.FirstOrDefault().IngredientUnit,
28         Quantity = iig.Sum(i => i.Quantity),
29         Ingredient = iig.FirstOrDefault().Ingredient,
30         ExpirationDate = iig.FirstOrDefault().ExpirationDate,
31         PurchaseDate = iig.FirstOrDefault().PurchaseDate,
32         User = iig.FirstOrDefault().User,
33         UserID = iig.FirstOrDefault().UserID
34     };
35
36 //Same as last query except it's a list not a IQueryable.
37 public List<InventoryListGroupedByQuantity> InventoryList = (from ii in App.db.InventoryIngredients
38     join i in App.db.Ingredients on ii.IngredientID equals i.ID
39     where ii.UserID == App.CurrentUser.ID
40     group ii by ii.IngredientID into iig
41     select new InventoryListGroupedByQuantity()
42     {
43         IngredientID = iig.FirstOrDefault().IngredientID,
44         Unit = iig.FirstOrDefault().IngredientUnit,
45         Quantity = iig.Sum(i => i.Quantity),
46         Ingredient = iig.FirstOrDefault().Ingredient,
47         ExpirationDate = iig.FirstOrDefault().ExpirationDate,
48         PurchaseDate = iig.FirstOrDefault().PurchaseDate,
49         User = iig.FirstOrDefault().User,
50         UserID = iig.FirstOrDefault().UserID
51     }).ToList();
52
53 //A query that get all the ingredients from earlier meals, and group them, where the IngredientCount is the numbers of times the specific ingredient has been used.
54 public List<LastMeal> IngredientsFromLastMeals = (from meals in App.db.Meals
55     join ri in App.db.RecipeIngredients on meals.RecipeID equals ri.RecipeID
56     join i in App.db.Ingredients on ri.IngredientID equals i.ID
57     where meals.UserID == App.CurrentUser.ID && meals.Date <= DateTime.Now
58     group i by i.ID into igrouped
59     select new LastMeal()
60     {
61         IngredientID = igrouped.FirstOrDefault().ID,
62         IngredientCount = igrouped.Count()
63     }).ToList();
64
65 //A query that gets a list recipes where a blacklisted ingredient is used.
66 public List<int> blacklist = (from bi in App.db.BlacklistIngredients
67     join ri in App.db.RecipeIngredients on bi.IngredientID equals ri.IngredientID
68     where bi.UserID == App.CurrentUser.ID
69     group ri by ri.RecipeID into ri
70     select ri.FirstOrDefault().RecipeID).ToList();
71
72 //get a list of graylisted ingredients.
73 public List<Graylist> graylist = (from gi in App.db.GraylistIngredients
74     join i in App.db.Ingredients on gi.IngredientID equals i.ID
75     where gi.UserID == App.CurrentUser.ID
76     group gi by gi.IngredientID into gi
77     select new Graylist()
78     {
79         Ingredient = gi.FirstOrDefault().Ingredient,
80         rating = (gi.Sum(r => r.IngredientValue) / gi.Count())
81     }).ToList();
82 #endregion

```

26

Search by recipes

Food Planner

DS301E14

Intro

Problem statement &
Information Gathering

Quality Assurance

Unit Testing

Usability Testing

Reflection

Code Examples

24

```

83 | #region Methods
84 | #region Search
85 | public IQueryable<IGrouping<int, Result>> search(List<int> recipeIDs)
86 | {
87 |     //IQueryable that finds all the ingredients that belong to the found recipes and how much of the found ingredient.
88 |     IQueryable<IGrouping<int, Result>> recipeIngredients = from ri in App.db.RecipeIngredients
89 |                                                           join i in App.db.Ingredients on ri.IngredientID equals i.ID
90 |                                                           join r in App.db.Recipes on ri.RecipeID equals r.ID
91 |                                                           where recipeIDs.Contains(ri.RecipeID)
92 |                                                           group new Result()
93 |                                                           {
94 |                     recipe = ri.Recipe,
95 |                     ingredient = ri.Ingredient,
96 |                     quantity = ri.Quantity
97 |                 } by ri.RecipeID;
98 |
99 |     return recipeIngredients;
100 | }

```

26

Sorting of results

Food Planner

DS301E14

Intro

Problem statement &
Information Gathering

Quality Assurance

Unit Testing

Usability Testing

Reflection

Code Examples

25

```

101 // @Override
102 public ObservableCollection<SearchResults> addValuesToSearch(IDictionary<int, Result>> userInput, List<string> searchKeywords = null)
103 {
104     //Create ObservableCollection which allows for updates in the view.
105     ObservableCollection<SearchResults> result = new ObservableCollection<SearchResults>();
106
107     //Traverse through all found ingredients and add them to the recipe
108     foreach (IDictionary<int, Result> r1 in userInput)
109     {
110         Recipe recipe = r1.FirstOrDefault().recipe;
111
112         //Initialize the searchResult that is to be shown
113         SearchResults searchResult = new SearchResults(recipe);
114
115         //Make sure that searchKeywords is set, if null assume that it's a non-search that need to have added value.
116         if (searchKeywords != null)
117         {
118             //Check that the keyword is found in the recipe, it's case sensitive
119             if (searchKeywords.Any(s => recipe.Title.ToLower().Contains(s.ToLower())))
120             {
121                 searchResult.keywordMatch++;
122             }
123
124             //Traverse through all results which is ingredients
125             foreach (Result res in r1)
126             {
127                 //Add the ingredient to the recipe
128                 searchResult.addIngredient(res.ingredient);
129
130                 //Check if the ingredient is found in the users' inventory
131                 if (InventoryList.Where(i1 => i1.IngredientID == res.IngredientID).Count() != 0)
132                 {
133                     //If found check if it's a partial match or full match
134                     //full match means that all of the ingredient is found in the inventory
135                     if (InventoryList.Where(i1 => i1.IngredientID == res.IngredientID).First().Quantity >= res.quantity)
136                     {
137                         searchResult.fullMatch++;
138                     }
139                     else
140                     {
141                         //If only partial match, and the partial match percentage to the partialMatch
142                         searchResult.partialMatch += InventoryList.Where(i1 => i1.IngredientID == res.IngredientID).First().Quantity / res.quantity;
143                     }
144                 }
145             }
146         }
147     }
148 }

```

26

Sorting of results

Food Planner

DS301E14

Intro

Problem statement &
Information Gathering

Quality Assurance

Unit Testing

Usability Testing

Reflection

Code Examples

26

```

143     }
144     //Make sure that searchKeywords is set, if null assume that it's a non-search that need to have added value.
145     if (searchKeywords != null)
146     {
147         //check that the keyword is found in the ingredient, it's case sensitive
148         if (searchKeywords.Any(s => res.ingredient.Name.ToLower().Contains(s.ToLower())))
149         {
150             searchResult.keywordMatch++;
151         }
152     }
153 }
154
155 //Check if the ingredient is used in previous meals.
156 if (ingredientsFromLastMeals.Where(ifm => ifm.ingredientID == res.ingredient.ID).Count() != 0)
157 {
158     //If found add the amount of times the ingredient has been used to the prevIngredients
159     searchResult.prevIngredients += ingredientsFromLastMeals.Where(ifm => ifm.ingredientID == res.ingredient.ID).Single().ingredientCount;
160 }
161
162 //Check if the ingredient is found in the gray list, if it is add the value to setRating
163 if (grayList.Where(gl => gl.ingredientID == res.ingredient.ID).Count() != 0)
164 {
165     searchResult.setRating = grayList.Where(gl => gl.ingredientID == res.ingredient.ID).SingleOrDefault().rating;
166 }
167 else
168 {
169     //50 is the default value of nonrated items
170     searchResult.setRating = 50;
171 }
172 }
173
174 //add the SearchResults to the ObservableCollection of SearchResults
175 result.Add(searchResult);
176 }
177
178 //return the ObservableCollection ordered.
179 return new ObservableCollection<SearchResults>(result.OrderByDescending(res => res.percentageFullMatch).ThenByDescending(res => res.percentagePartialMatch).ThenByDescending(res => res.setRating));
180 }
181 }
182 }
183 }
184 }
185 }

```