EDAF20: Project Description
Cookie Production and Delivery

Department of Computer Science
Lund Institute of Technology

January 15, 2015

The programming project is compulsory. You are to work in groups of two or three people, and it is up to you to find your group partners. If you have difficulties finding a group partner, send an e-mail to Per.Andersson@cs.lth.se.

The project consists of three parts: database modeling, database implementation, and development of (parts of) a user interface to the database.

There will be two deliverables during the project. The first deliverable contains the database design, which will be reviewed during exercise session 6. The second deliverable is the final report and implementation.

All deliverables are to be emailed to edaf20@cs.lth.se. The subject of the email must be ”Krusty by user1 and user2”. For student groups with 1 and 3 students use the title ”Krusty by user1” or ”Krusty by user1 and user2 and user3”. Replace user1 and user2 with the correct user names. It is the same as listed for the lab groups.

Questions about the project can be directed to Per.Andersson@cs.lth.se.

1 Assignment

1.1 Introduction

In this project you will model and implement a database for the supervision of production and delivery of cookies produced by the company Krusty Kookies Sweden AB. The company and its routines for production and delivery are presented in section 2. The requirements for the database and for the interface to the database are presented in section 3. In section 4 you will find modifications of the requirements, suitable for the pilot implementation that you are to perform.

Notice that the text in sections 2 and 3 is written by the company. This may have had the consequences that some of the requirements are in the section containing the company presentation, and also that some text in the requirements section is company promotional material. An analysis of the requirements is part of the assignment. You should make reasonable assumptions where the description is unclear.

1.2 Modeling

Produce a complete database description with an E/R diagram and a relational model.

1.3 Review a database design

During exercise session 6 you are reviewing another groups model, and that group will review your.

1.4 Implementation of the Database

Implement the database. Document and justify any deviations from the model.

1.5 Implementation of the User Interface

You are to implement a graphical user interface to the database (only ”program 2” of the programs mentioned in section 4).

- All requirements must be addressed.
- The implementation must be stable. The system should produce reasonable results for arbitrary input.
- The user interface must be self-explanatory and easy to use.

1.6 Alternative Implementations

The recommended alternative for implementation is that you use the department's MySQL system and write a Java application that communicates with the database with JDBC calls and uses Swing for the graphical user interface, as you did in lab 3. You will probably be able to use the GUI classes that you used in lab 3, suitably modified.

However, we do not require that you use this method of implementation. You may use a different database system (PostgreSQL, . . .), you may use a different programming language (C, C++, Visual Basic, . . .), you may use another form of interface (developed in JBuilder, web pages with Perl, PHP, servlets, JSP, ASP, . . .). If you choose a different implementation than java and swing the support will be limited.

Two requirements are non-negotiable:

1. your database must use ”normal” SQL that you write by yourselves (so Microsoft Access is not allowed), and
2. it must be possible to test your programs at a computer, connected to the Internet, in the campus. As an alternative you may develop the entire system on your own computer, but then you must bring a laptop computer to the department to demonstrate your program.

If you plan to use some exotic combination of database and interface implementation, and are not sure how to interpret the requirements, we recommend that you ask *before* you start the programming work.

1.7 Final Report

The final report must be well structured, complete, easy to understand, well formulated, etc. The report may be written in Swedish or in English. The intended reder is a person that knows Java, swing and database design, but have not read the assignment and don not know of Krusty Kookies Sweden AB. Contents (you may omit from or add to this list if you can justify your changes):

1. A cover sheet with the name of the project, your names, education programs, and e-mail addresses.
2. An abstract in English.
3. An introduction (what the project is about, etc.).
4. The extent of your solution (which requirements that you fulfill in your solution).
5. Limitations, e.g., requirements that you have not implemented. Known bugs in the program. Such deviations must be justified.

6. An outline of your system (which database manager you use, which programs you have written, how the programs communicate with the database, etc.).
7. An E/R diagram which describes the system.
8. Relations. Indicate primary keys, possibly secondary keys, and foreign keys. You must show that the relations are normalized according to your chosen normal form (if a relation ”obviously” is in BCNF you may say so, provided that you justify your statement). If a relation is in a lower normal form than BCNF, you must justify this choice.
9. Examples of SQL statements for queries or updates. Explain and show how you update the raw materials storage when a pallet of cookies is produced.
10. An example that shows how the program communicates with the database, e.g., a Java method containing JDBC calls.
11. A description of the implementation of the graphical user interface (programming language, classes, relationships between classes, etc.)
12. Conclusions. Requirements that you fulfill, requirements that you did not succeed to fulfill, etc. If you have found that the system ought to have more features you should indicate this here.
13. Comments (for the department) concerning the project: good and bad features, comments on the size of the assignment, should something be changed for next year etc. This could also be added to the CEQ-course evaluation form.
14. A user manual. The target group for the manual is formed of teachers who know the assignment well and also know the course well. The manual is preferably task oriented, i.e. how to register a baked pallet, how to block cookies not satisfying the quality requirements, et.c. If everything in the program is self-explanatory, screen shots and the title of the task the views handles are sufficient.

Most of the items above should be placed in separate chapters, but some may be merged to one chapter. The focus should be to structure the report for the best readability, not following a strict form. Long program listings (java/sql) is best to place in apendixes.

2 Krusty Kookies Sweden AB

2.1 Company Presentation

Krusty Kookies Sweden AB is the new name for the company earlier called Småkakor AB. We changed our name when we were taken over by the multi-national company Krusty Kookies. Our business idea is, as it always has been, to provide our customers with a broad range of high-quality cookies, and the unique production and delivery concept developed by Krusty Kookies will allow us to meet our goals.

Krusty Kookies gives us the opportunity for new investments that will increase customer satisfaction with our products. We will introduce a new production line in the bakery, and the routines for storage management will be updated. This will increase our production capacity and make a diversification of our product range possible, and will enable us to reach new groups of customers.

2.2 A Computerized System for Production and Delivery

Earlier, we used manual routines for planning and follow-up of production and delivery. This worked well in the small scale in which we used to work, but since the scope of our operations has widened we plan to use computerized routines. One of the basic ideas of Krusty Kookies is to take geographic and national differences into account, so the system will be developed locally in Sweden.

The computerized system must be flexible and be able to meet our demands on usability and reliability. In 2.3-2.7 we describe how production and delivery will function in our new production plants. In section 3 you will find detailed requirements on the system.

In the prototype implementation the simulated production and delivery data found in appendices A and B should be used.

2.3 Production and Packaging

Our raw materials warehouse contains all ingredients that are necessary for our different products. We will continue to use our well-proven manual routines to order raw materials, so it can be assumed that all necessary ingredients are present in sufficient quantities for the production of the desired products.

Cookies are baked in our new ovens, following our traditional and proven recipes (appendix A). Each type of cookie is baked in large quantities. The cookies are quick frozen and packaged in bags, with 15 cookies in each bag. The bags are put in boxes, with 10 bags in each box.

The boxes are stacked on loading pallets, which are shrink-wrapped in plastic and labeled with bar code labels. Each pallet contains 36 boxes, all containing the same product. It takes 10 minutes to produce and load a pallet of cookies. This impressive speed is due to our new mixers and ovens, which use the latest advances in bakery technology.

Appendix C shows a sketch depicting the production process, and also storage and delivery.

2.4 Storage

When a pallet is produced and labeled, it is transported to the deep-freeze storeroom. When the pallet reaches the storeroom the pallet label is read by a bar code reader.

2.5 Orders

Customers place their orders by telephone. All customers (appendix B) must be registered in our database (we have special routines for checking and registering customers). Orders are of the type ”send 10 pallets of Tango cookies and 6 pallets of Berliners on Thursday next week”. We only deliver full pallets.

2.6 Loading

Pallets are transported from the deep-freeze storeroom via a loading ramp to the freezer trucks. Each truck loads 60 pallets. The entry to the loading ramp contains a bar code reader which reads the pallet label. Pallets must be loaded in production date order. The time to load a pallet is approximately one minute.

When the truck is fully loaded, the driver receives a loading bill containing customer names, addresses, and the number of pallets of each product that is to be delivered to each customer. A transport may contain deliveries intended for different customers.

We deliver only to wholesale customers (appendix B). We again point out that this data is simulated — the number of customers will increase considerably in the near future. We are also planning to expand our market to all of Scandinavia.

In the future, the data from the delivery system will be integrated with our existing invoicing system.

2.7 Quality Assessment

In order to guarantee high product quality we continuously take random samples among the products. The samples are analyzed in our laboratory. If a sample does not meet our quality standards all pallets containing that product, that have been produced during a specific time, are *blocked*. A blocked pallet may not be delivered to customers.

3 System Requirements

3.1 Production

A pallet is considered to be produced when the pallet label is read at the entrance to the deepfreeze storage. The pallet number, product name, and date and time of production is registered in the database. The pallet number is unique.

At any time, we must be able to check how many pallets of a product that have been produced during a specific time.

3.2 Raw Materials

When a pallet is produced, the raw materials storage must be updated. We must be able to check the amount in store of each ingredient, and to see when, and how much of, an ingredient was last delivered into storage.

3.3 Recipes

We need an interface to the collection of recipes (appendix A), where we can study and update recipes. We also need a facility for entering new recipes. We do not change recipes during production.

3.4 Produced Pallets

As we mentioned earlier, pallets in the deep-freeze storage may be blocked. An order to block a pallet will always come before the pallet has been delivered. This is due to the new investments in our laboratory, where the analysis process is completely automated.

We must be able to trace each pallet. For instance, we need to see all information about a pallet with a given number (the contents of the pallet, the location of the pallet, if the pallet is delivered and in that case to whom, etc.). We must also see which pallets contain a certain product, and which pallets have been produced during a certain time interval.

Blocked products are of special interest. We need to find out which products that are blocked, and also which pallets that contain a certain blocked product.

Finally, we must be able to check which pallets that have been delivered to a given customer, and the date and time of delivery.

3.5 Orders

Orders must be registered in the database. For production planning purposes, we must have a way to see all orders that are supposed to be delivered during a specific time period.

3.6 Delivery

Before pallets are loaded into the freezer trucks, a loading order is created. The order contains information regarding the customers and the number of pallets to be delivered.

When pallets are taken out of deep-freeze storage the pallet label is read. When the truck is loaded, the driver receives a loading bill (identical to the loading order, but contains a field where the customer can acknowledge reception of the delivery). The loading bill data need not be saved in the database.

When the loading bill has been printed, the data regarding delivered pallets must be updated with customer data and date of delivery.

4 Introductory Study

Here, the results of an introductory study of the project are given. We have studied the requirements specification and made decisions regarding the prototype implementation:

- We must be able to test the system without connecting it to the factory. For instance, we do not have access to the bar code readers that inform us about pallet production or pallet delivery. Instead, we introduce a screen in the user interface where we simulate pallet production, and another screen where we simulate pallet delivery. These screens will be removed from the production version of the system. The decision means that we can ignore all information about production and delivery speed.
- The database must be correct and we must be able to check all different cases, so we implement the entire database.
- The requirements specification mentions *the* program that interfaces to the database, but contacts with the company have informed us that different departments will use different parts of the program. Therefore, the system will be better if each department only can see the part of the database that concerns them. We will write three different programs that use the same database:
  1. One program that handles everything that concerns raw materials and recipes.
  2. One program that handles everything that concerns production[1], blocking and searching of pallets. The test version of

[1] this includes updating the raw materials storage when a pallet of cookies is produced

this program will incorporate the screen where pallet production is simulated.

3. One program that handles everything that concerns orders and deliveries.

- In the prototype implementation we only implement the program mentioned under item 2. When we show the system to the customer, we make sure that the database contains relevant data so we can demonstrate all of the functionality (we will use `mysql` or a similar client program to enter the data).

5 Submissions

For the project there are two submissions. The first containing the database model and the second the final report and the finished system. All hand ins is done by email. Send all files as attachments to the email. Send the email to edaf20@cs.lth.se. The subject line of the email must be (for a group with two people): `Krusty by user1 and user2`. Write your full names in the email body and on the cover of the report.

5.1 Database model

The latest date for submission is on the course home page. Your submission will be reviewed during excercise session 6. When you are ready to submit, create a pdf containing:

1. A title indicating the content, for example "Krusty, database design", and the name of all group participants.
2. A picture of your ER-model.
3. The relational model derived from your ER-model.
4. A brief description of the the database operations involving (SQL, swedish, or english):
   (a) Baking a pallet of cookies.
   (b) Receiving an order from a customer.
   (c) Delivering an order to a customer.

Send the email to edaf20@cs.lth.se containing the pdf and write your names in the body of the email.

5.2 Final report

The latest date for submission is on the course home page. Submit by email to edaf20@cs.lth.se:

1. The title of the email must be `Krusty by user1 and user2`. Replace user1 and user2 by your user ids.
2. Write all group participants names in the body of the email.
3. Produce a *pdf* file of your final report. The reports content is detailed in 1.7.
4. A file with SQL statements to create all tables, views, stored procedures, and other database elements. Also add SQL statements to create the initial content of the tables, i.e. the same as makelab1.sql you used for the first lab. This is easiest done using the *Data Export* function of MySQL Workbench. You find the function in the *server* menu.
5. All source code files for your program (.java) in an archive (*zip* or *tar.gz* format). Include any other files that are needed to test your system. If you are using large third party software/libraries, such as tomcat, do not include these. Instead provide a link to where they can be downloaded.
6. Instructions on how to run your program, i.e. which class contains the `main()` method. Either in a *README* text file, or in the body of the email.

Appendix A: Products

The table below contains recipes for some of our traditional products. Each recipe is for 100 cookies.

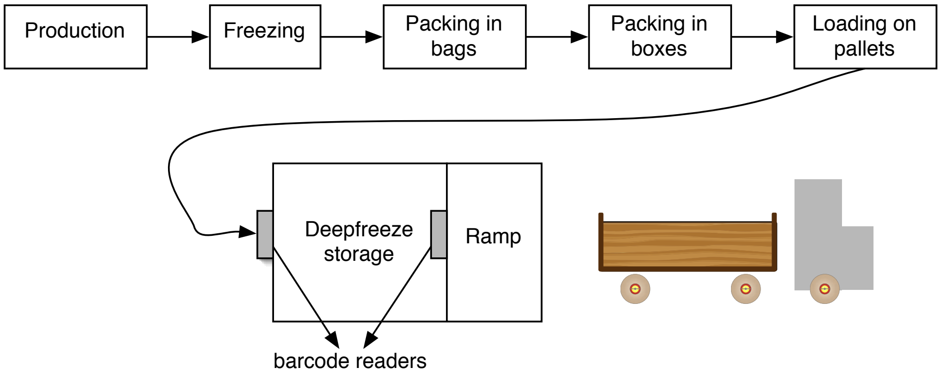
Nut ring	Flour	450 g
	Butter	450 g
	Icing sugar	190 g
	Roasted, chopped nuts	225 g
Nut cookie	Fine-ground nuts	750 g
	Ground, roasted nuts	625 g
	Bread crumbs	125 g
	Sugar	375 g
	Egg whites	3,5 dl
	Chocolate	50 g
Amneris	Marzipan	750 g
	Butter	250 g
	Eggs	250 g
	Potato starch	25 g
	Wheat flour	25 g
Tango	Butter	200 g
	Sugar	250 g
	Flour	300 g
	Sodium bicarbonate	4 g
	Vanilla	2 g
Almond delight	Butter	400 g
	Sugar	270 g
	Chopped almonds	279 g
	Flour	400 g
	Cinnamon	10 g
Berliner	Flour	350 g
	Butter	250 g
	Icing sugar	100 g
	Eggs	50 g
	Vanilla sugar	5 g
	Chocolate	50 g

Appendix B: Customers

The table shows some of our customers (wholesalers).

Customer	Address
Finkakor AB	Helsingborg
Småbröd AB	Malmö
Kaffebröd AB	Landskrona
Bjudkakor AB	Ystad
Kalaskakor AB	Trelleborg
Partykakor AB	Kristianstad
Gästkakor AB	Hässleholm
Skånekakor AB	Perstorp

Appendix C: Production, Storage, and Delivery


Production
Freezing
Packing in bags
Packing in boxes
Loading on pallets
Deepfreeze storage
Ramp
barcode readers