|  |  |
| --- | --- |
| **Project – External Documentation** |  |
| <Course Code>  <Course Name> |
| <Odd/Even/Compact> Semester Year <9999/9999> |

**Raamen**

**Dokumentasi Project PSD**

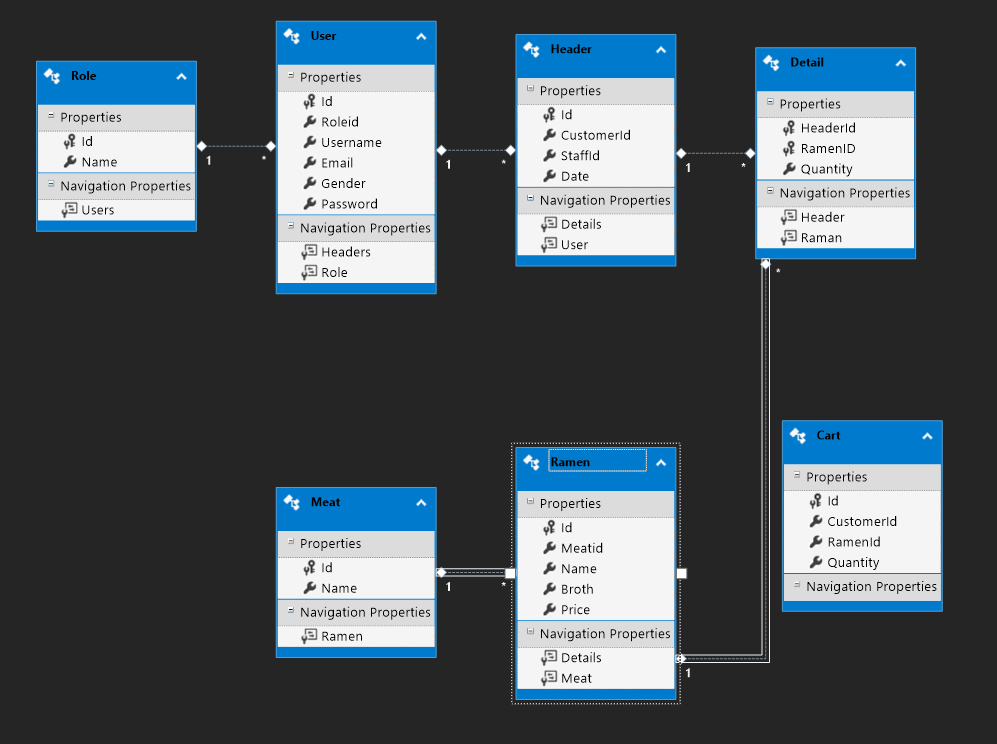
**Kelompok:**

2502036672 - Raien Ronariv

2502044264 - Bryan Amadeus

2502016475 - Haliza Syafa Oktaviani

**Model:**



**Query DB ada di txt**

**Contoh Database:**

**A screenshot of a computer

Description automatically generated with medium confidence**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated with medium confidence**

**A screenshot of a computer menu

Description automatically generated with medium confidence**

**A screenshot of a computer

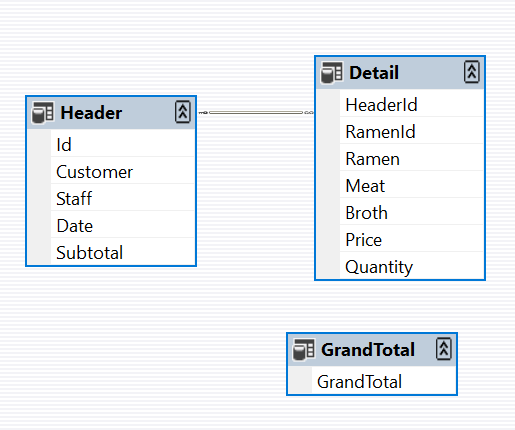
Description automatically generated with medium confidence**

A screenshot of a computer

Description automatically generated

**Report:**

**Dataset:**



**Crystal Report:**

A screenshot of a computer

Description automatically generated with medium confidence

Detail

A screenshot of a computer

Description automatically generated

**Controller:**

**UserController.cs**

Untuk validasi saat createmember dan update member

Untuk updatesession setelah update member

Validasi cek alphabet

Validasi login user

Untuk menghubungkan method ke views dari handler

**RamenController.cs**

Validasi sebelum insert ramen, update ramen.

Untuk menghubungkan method ke views dari handler

**TransactionController.cs**

Untuk menghubungkan method ke views dari handler

**CartController.cs**

Untuk menghubungkan method ke views dari handler

**Factory:**

**UserFactory.cs**

Untuk membuat database user yang baru

**CartFactory.cs**

Untuk membuat database cart yang baru

**TransactionFactory.cs**

Untuk membuat transaction yang baru detail dan header

**RamenFactory.cs**

Untuk membuat database ramen yang baru

**Handler:**

Handle method repository ke controller

**CartHandler.cs**

Menghubungkan ke CartRepository

**TransactionHandler.cs**

Menghubungkan ke TransactionRepository

**RamenHandler.cs**

Menghubungkan ke RamenRepository

**UserHandler.cs**

Menghubungkan ke UserRepository

**Repository:**

**CartRepository:**Mengatur database cart

Untuk mendapatkan list cart, dan menambahkan cart yang baru, menghubungkan ke factory

Dan untuk menghapus cart

**TransactionRepository:**

Mengatur transaction

Mengambil list transaction, menghubungkan ke factory

Juga untuk handle transaction

**UserRepository:**

Mengatur user, cek email yang unique, cek credential login, menghubungkan ke factory

Mengambil list user, update db User

**RamenRepository:**

Mengatur ramen, menghubungkan ke factory untuk create ramen.

Delete ramen, dan untuk update db ramen, dan mendapatkan list ramen atau ramen sendiri, juga untuk mendapatkan list meat.

**Views:**

**Login**

Untuk login ke website

Cookie jika klik remember me akan tersimpan,

LoginButton untuk login dan validasi ke UserController

Validasi dilakukan di controller

**Register**

Untuk register akun,

RegisterBtn untuk melakukan validasi register dan register account.

Setelah validasi akan create user melalui UserController, UserHandler, UserRepository dan UserFactory.

Validasi dilakukan di controller

**Site1.Master**

Untuk master page,

Berisikan navbar sesuai requirement Role

Semua user memiliki menu profile,home dan button logout

User punya order ramen, history

Staff punya manage ramen, orderQueue

Admin punya manageramen, orderqueue,history dan report.

Master page digunakan semua webform dalam views.

**Home**

Home berisikan data-data user bagi Staff

Dan untuk admin berisikan data staff.

Data ditampilkan dalam gridview yang didapatkan dari list db user sesuai rolenya.

**Profile**

Dapat melakukan update profile dengan validasi ke UserController, user dapat update username, email, gender.

Untuk password untuk validasi harus sama dengan password sebelumnya.

Hanya dapat dilakukan user, jika admin atau staff update profile akan langsung redirect ke home.

**History**

Berisi history transaction

Jika nama staff bukan staff berarti merupakan unhandled

Untuk role user hanya menampilkan yang telah dilakukan user

Untuk admin menampilkan semuanya

Untuk melihat detailnya dapat mengklik detail lalu akan redirect ke transactiondetail menggunakan query string.

**TransactionDetail**

Mengambil id transaction dari query string link. Dan mengambil semua detail yang memiliki id yang sama dengan header.

Memiliki tombol back untuk Kembali ke history

**OrderRamen**

Berisikan data ramen, untuk order klik kotak ramennya dan akan masuk ke cart. Jika admin dan staff yang klik maka tidak akan terjadi apa-apa.

Di cart berisikan data ramen yang belum dibuy.

Ada 2 tombol di bawah cart yaitu buy untuk membeli ramen dan membuat unhandled transaction baru,

Tombol clear akan menghapus cart yang ada.

**TransactionQueue**

Untuk ke page ini dapat menggunakan tombol orderqueue

Berisikan data unhandled transaction, ini akan ditampilkan hanya untuk staff dan admin. Staff/admin dapat handle yang dipilih saja atau handle smua unhandled dengan tombol Handle All

**ManageRamen**

Menu ini berisi list ramen yang ada, user tidak bisa access menu ini, staff/admin dapat edit atau delete ramen.

Delete ramen maka akan menghapus ramen yang dipilih.

Jika edit maka akan redirect ke UpdateRamen dengan query string id.

**InsertRamen**

Untuk membuat ramen baru, dengan validasi RamenController.

Jika insertRamen maka akan create ramen yang baru.

**UpdateRamen**

Data-data ramen dapat diupdate disini, yang sesuai dengan validasi dari ramenController.

Jika sudah sesuai maka akan update database.

**TransactionReport**

Disini berisikan report transaction yang ada. Page ini hanya bisa dibuka admin.

Semua detail ramen diperlihatkan disini, beserta total harga ramen yang dibeli, subtotal dan grandtotal.

Subtotal untuk setiap header dan grandtotal untuk semua transaksi.