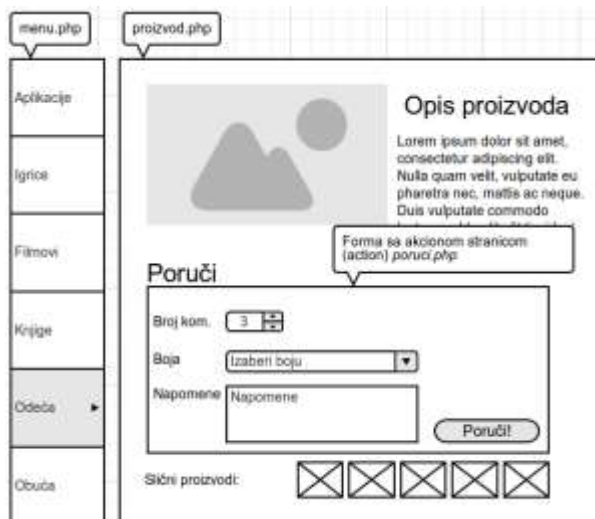


Трајање колоквијума: 2 сата

1. [6] а) Направити мапу дијалога за дати дијалог претраге са сајта ЕТФ-а.
- б) Описати користећи језик *Gherkin* неке сценарије претраге из наведеног дијалога.

2. [8] а) На слици је дат приказ структуре HTML странице коју генерише скрипта *kupi.php*. Мени (слева) генерише скрипта *menu.php*, а централни део генерише скрипта *produkt.php*. Одељак слични производи саджи везе (*eng. link*) ка страницама за куповину сличних производа. Притиском на везу, приказује се страница за куповину посматраног производа. Веза као податак садржи идентификатор производа (<http://radnja.com/kupi.php?id=33>). Форма за поручивање се подноси (*eng. submit*) скрипти *zavrshi\_kupovinu.php*. Након успешне куповине, скрипта приказује извештај о куповини. Користећи UML и WAE (*WebApp Extension*) моделовање, нацртати дијаграм који приказује све клијентске и серверске странице, и њихове компоненте, као и односе међу њима.



б) За случај употребе описан у наставку, дати одговарајући дијаграм секвенце. Претпоставити да су све провере успешно остварене. Не треба приказати интеракцију са базом података.

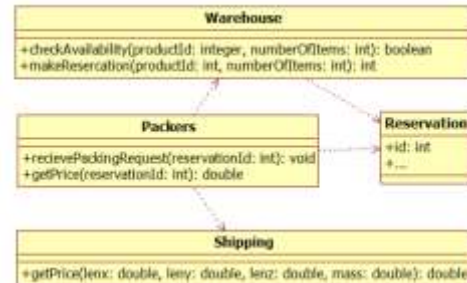
Корисник подноси поруџбину коју је потребно остварити

1. Корисник уноси потребне податке: број јединица (комада) производа, боја и напомене. Притиском на дугме „*Poruči*“ шаље захтев за паковањем и слањем одабраног производа.
2. Уколико у складишту постоји потребан број комада производа, они се резервишу, и пакерима се шаље налог за паковање и слање.

3. Дохвата се коначна цена паковања и слања за обрађену поруџбеницу и ти подаци се приказују кориснику.

Компоненте система (класе) које се користе при реализацији описаног ССУ, и функционалности које оне пружају, приказане на класном дијаграму поред, су:

- *Warehouse* - складиште. Остварује проверу стања робе (да ли производ са задатим идентификатором постоји у задатој количини), при чему враћа статус (*true/false*) и прави резервацију производа (задати број јединица производа са задатим идентификатором), при чему враћа идентификатор резервације;
- *Packers* - пакери. Остварују пријем захтева за паковањем, при чему им се доставља идентификатор резервације за коју би требало спаковати робу, и остварују проверу цене паковања и слања. Паковање се дешава у произвољном тренутку (не одмах по пријему захтева). Цена слања се добија преко *Shipping*.
- *Shipping* - слање. Остварују дохватање цене слања за пакет задатих димензија (висина, ширина, дужина) и масе, као и пријем захтева за слање спакованог пакета, задатог идентификатором.



3. [6] а) Написати PHP скрипте страница **kupi.php**, **proizvod.php** и **zavrshi\_kupovinu.php**, према изгледу слике и опису датом у другом задатку. Претпоставити да се подаци о производу дохватају из MySQL базе података из табеле Производ (Шифра\_Производа, Назив, ИД\_Категорије\_Производа, Опис, Слика, Количина\_На\_Лагеру, ИД\_боје). Уколико је потребно увести и друге потребне табеле. Слични производи генеришу се као низ 5 случајно одабраних производа, који припадају истој категорији. Потписи потребних метода налазе се у прилогу:

```

resource mysql_connect (string $server, string $username, string $password);
bool mysql_select_db (string $database_name)
mixed mysql_query (string $query)
int mysql_num_rows (resource $result)
array mysql_fetch_array(resource $result)
array mysql_fetch_assoc (resource $result)
int rand (void) - враћа број између 0 и 32767
int rand (int $min, int $max) - враћа број између MIN и MAX

```

б) Ради лакшег одржавања система, потребно је раздвојити логику и садржај од презентације података. Због тога је примењен *MVC (Model-View-Controller)* пројектни узорак. Дефинисати класе (и укратко их објаснити) и написати потписе метода које бисте увели, да део апликације који је описан буде MVC апликација.