TECHNISCHE UNIVERSITÄT DRESDEN

Department of Computer Science Institute for System Architecture, Chair for Computer Networks

Application Development for Mobile and Ubiquitous Computing

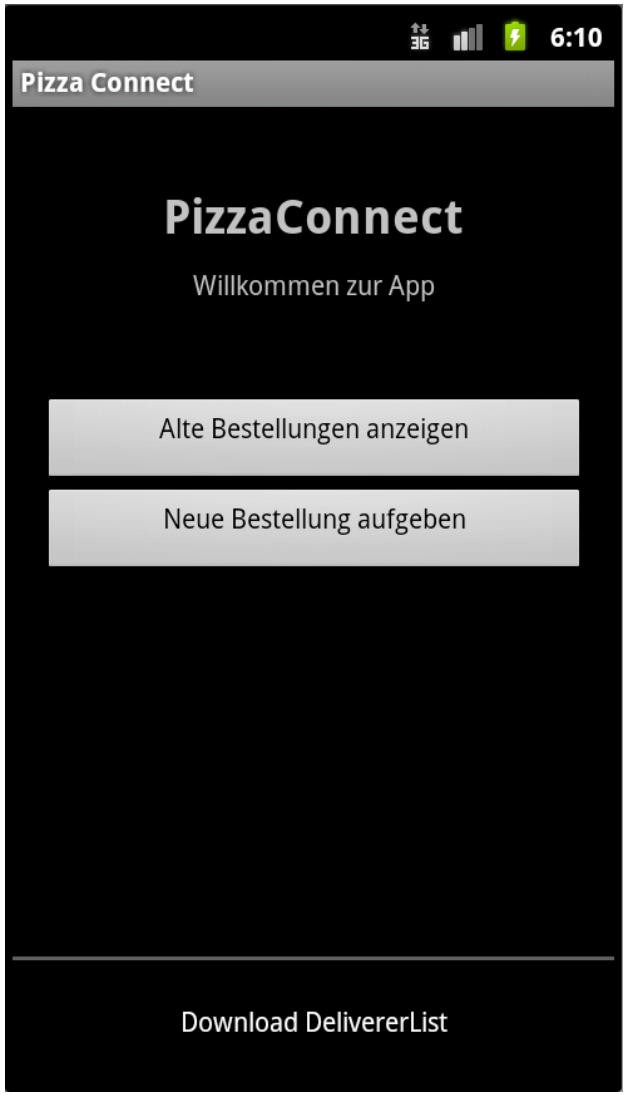
PizzaConnect
Final Presentation

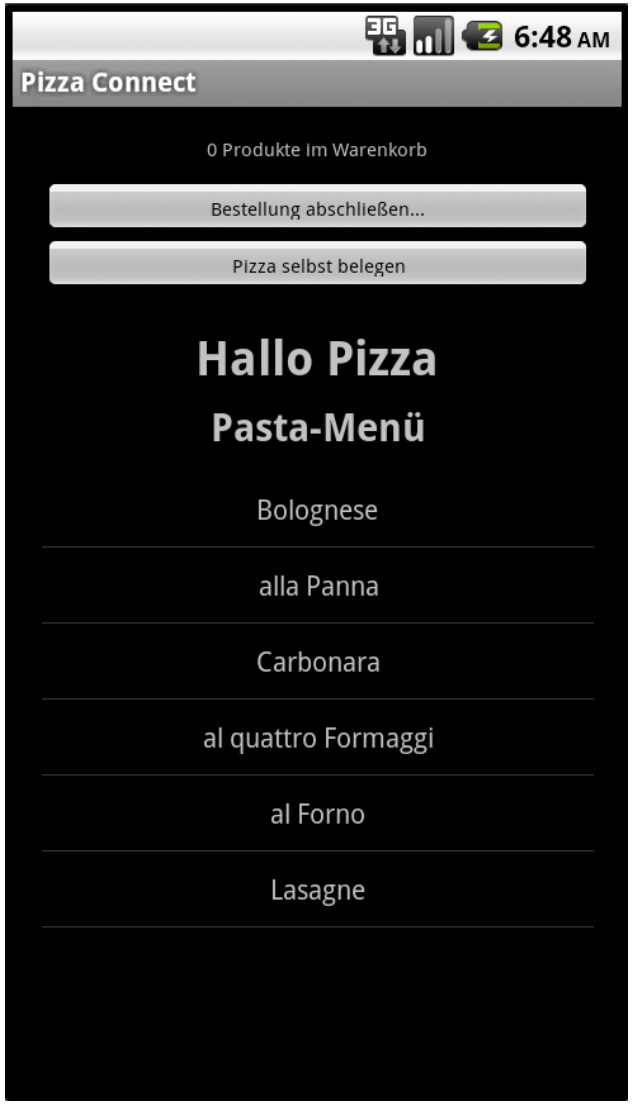
GroupNo. 3
Team: Thomas Walther &
Gregor Weimann

Concept

- wanted to develop an app for ordering at a delivery service
- user can choose his favorite delivery service and place his order
- can watch his older orders and can reorder them
- GPS usage to find location of user
- plating of pizza through drag and drop
- update of delivery services through webserver

Screenshots

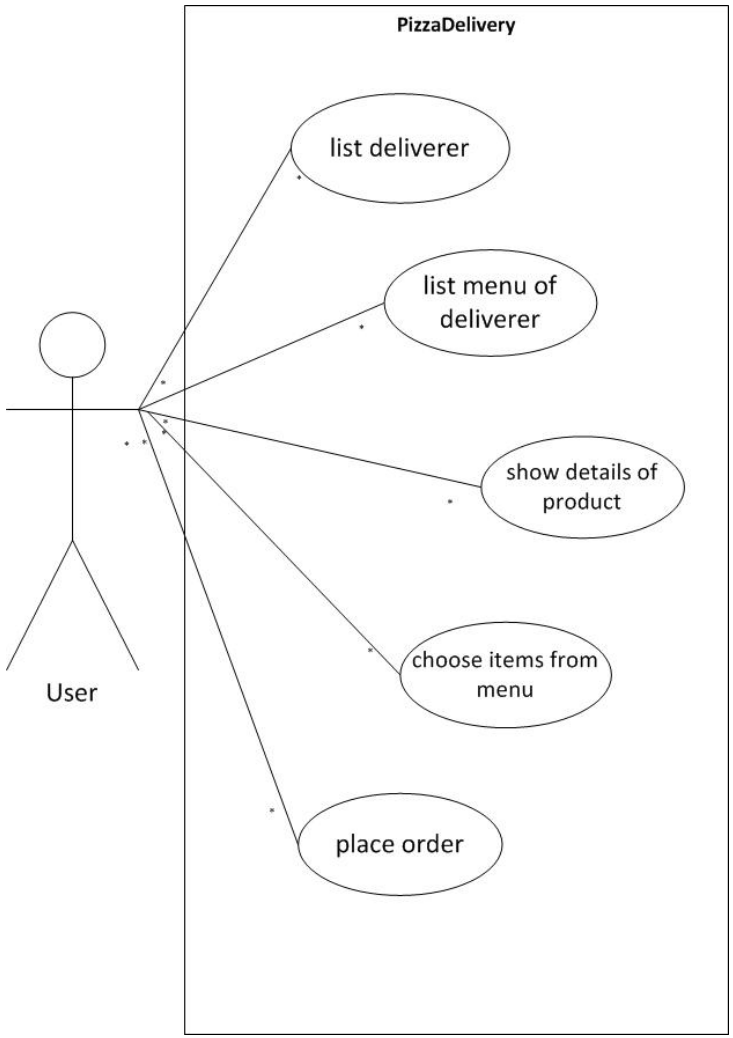




Screenshots

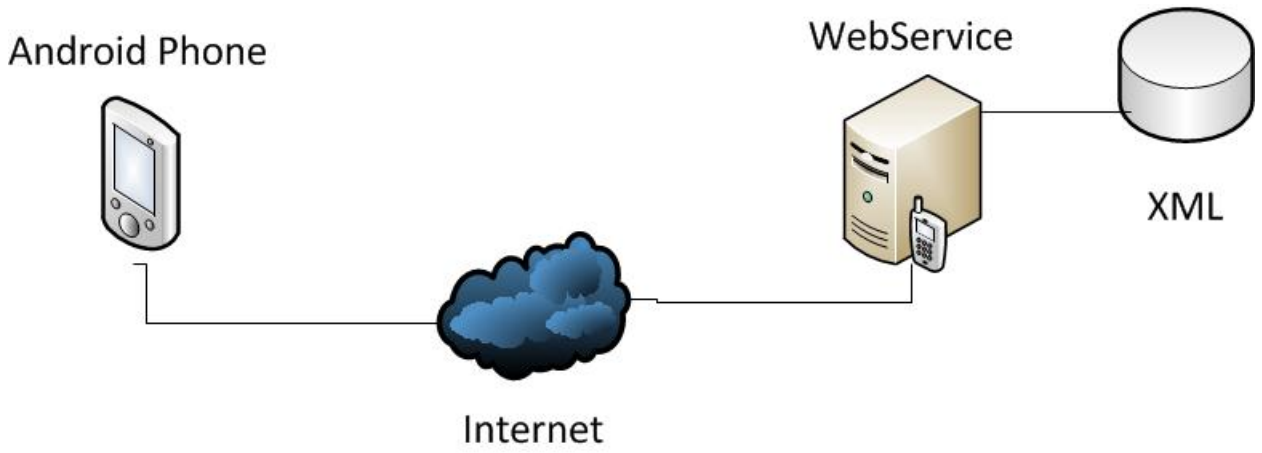
Use-Case

- first small use-case
- basic functions
- everything available
- wanted more



Final Architecture

- Client/Server architecture
- access webservice with REST
- client is android OS 2.2
- connection with WLAN/3G/GPRS
- reverse geopostioning through GPS



Implementation Details

```
if(delNodeItem.getNodeName().equals("Menu")) {
    // Menu
    Menu menu = new Menu();
    NodeList menuNodeList = delNodeItem.getChildNodes();
    for(int k=0;k<menuNodeList.getLength();k++) {
        Node menuNode = menuNodeList.item(k);
        if(menuNode.getNodeName().equals("Pizza")) {
            // Pizza
            Pizza pizza = new Pizza();
            NodeList pizzaNodeList = menuNode.getChildNodes();
            for(int l=0;l<pizzaNodeList.getLength();l++) {
                Node pizzaNode = pizzaNodeList.item(l);
                if(pizzaNode.getNodeName().equals("PizzaName")) {
                    pizza.setPizzaName(pizzaNode.getTextContent());
                }
                if(pizzaNode.getNodeName().equals("Toppings")) {
                    NodeList toppingsNodeList = pizzaNode.getChildNodes();
                    for(int x=0;x<toppingsNodeList.getLength();x++) {
                        Node toppingsNode = toppingsNodeList.item(x);
                        if(toppingsNode.getNodeName().equals("Topping")) {
                            pizza.getToppings().add(toppingsNode.getTextContent());
                        }
                    }
                }
                if(pizzaNode.getNodeName().equals("Description")) {
                    pizza.setDescription(pizzaNode.getTextContent());
                }
                if(pizzaNode.getNodeName().equals("Price")) {
                    pizza.setPrice(Double.valueOf(pizzaNode.getTextContent()));
                }
```

Implementation Details

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema elementFormDefault="unqualified" attributeFormDefault="unqualified" xmlns:xsd="http://www.w3.
    <xsd:element name="DeliverersList" type="DeliverersListItem"/>
    <xsd:complexType name="DeliverersListItem">
        <xsd:sequence>
            <xsd:element name="Deliverer" type="Deliverer" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="Deliverer">
        <xsd:sequence>
            <xsd:element name="DelivererName" type="xsd:string" maxOccurs="1" minOccurs="1"/>
            <xsd:element name="Menu" type="MenuItem" maxOccurs="unbounded" minOccurs="1"/>
            <xsd:element name="SelfPlate" type="xsd:boolean" maxOccurs="1" minOccurs="1"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="MenuItem">
        <xsd:sequence>
            <xsd:element name="Pizza" type="PizzaItem" maxOccurs="unbounded" minOccurs="0"/>
            <xsd:element name="Pasta" type="PastaItem" maxOccurs="unbounded" minOccurs="0"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="PizzaItem">
        <xsd:sequence>
            <xsd:element name="PizzaName" type="xsd:string" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="Toppings" type="ToppingItem" minOccurs="1" maxOccurs="unbounded"/>
            <xsd:element name="Description" type="xsd:string" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="Price" type="xsd:decimal" minOccurs="1" maxOccurs="1" />
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="PastaItem">
        <xsd:sequence>
            <xsd:element name="PastaName" type="xsd:string" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="Incredients" type="IncredientItem" minOccurs="1" maxOccurs="unbounded"/>
            <xsd:element name="Description" type="xsd:string" minOccurs="1" maxOccurs="1"/>
            <xsd:element name="Price" type="xsd:decimal" minOccurs="1" maxOccurs="1" />
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="ToppingItem">
        <xsd:sequence>
            <xsd:element name="Topping" type="xsd:string" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="IncredientItem">
        <xsd:sequence>
            <xsd:element name="Incredient" type="xsd:string" minOccurs="1" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
```

Limits and Contributions

- webserver need to grow for more functionality
  - picture delivery
  - mobile payment

    → power consuming

    → more space on device

    → security issue
- webserver interface for maintenance of the XML files
- website for access through browser

Tackled Mobile Problems

- bandwith limitations
  - small through XML-File on demand
  - saved on phone for later use
- usability
  - drag & drop ingredients with gestures
  - small screen adaptation
- power consumption
  - GPS only used a small amount of time
  - webservice connection only for getting list of deliverers and sending order to server

Pitfalls

- intended to have more functionality
- needed more time than planed for server-comunication and android to get familiar with
- continues working problem

Experience

- great expirence
- time-Consuming
- stay close to the time plan
- adopted new technology
- knowledge of server technolgy needed