

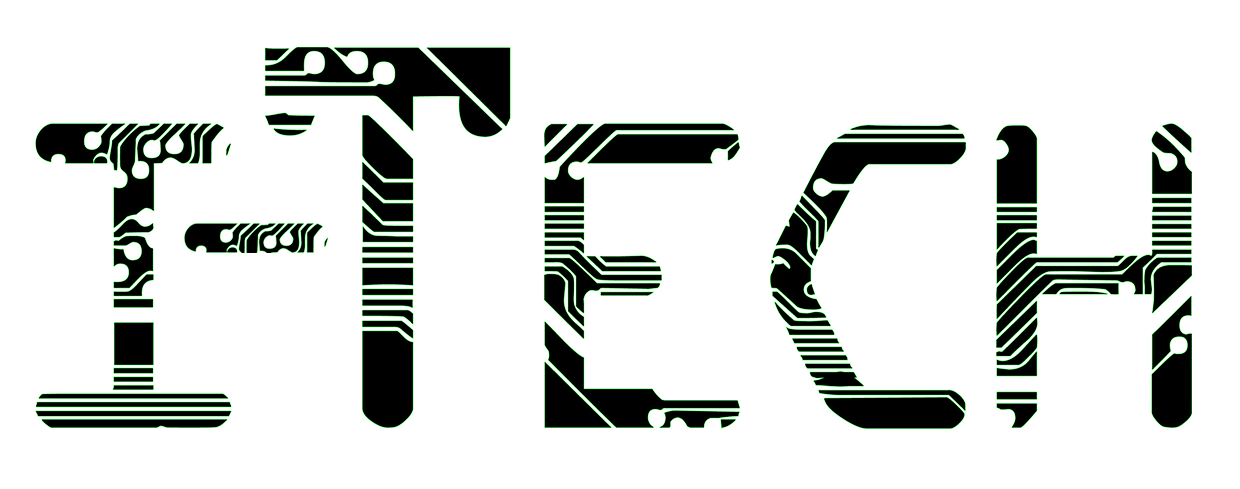
**Università degli Studi di Salerno**

**Anno Accademico 2016/2017**

**Corso di Ingegneria del Software**

**Requirements Analysis Document**

**V 4.0**

****

Sommario

Introduzione 4

Descrizione del sistema 4

Scopo del sistema 4

Obiettivi 4

Overview 4

Architettura del sistema corrente 4

Sistema proposto 5

Identificazione attori 5

Requisiti funzionali 5

Requisiti non funzionali 7

Modelli del sistema 8

Scenari amministratore 8

Scenari cliente 8

Casi d’uso 10

Use case utente generico: amministratore, cliente 10

Login 10

Use case 10

Use case diagram 11

Mockup - cliente 11

Mockup – Amministratore 15

Object Entity, Boundary, Control 19

Logout 19

Use case 19

Use case diagram 20

Mockup – Cliente 20

Mockup – Amministratore 21

Object Entity, Boundary, Control 24

Ricerca prodotto 24

Use case 24

Use case diagram 25

Mockup - cliente 25

Mockup – amministratore 29

Object Entity, Boundary, Control 33

Use case amministratore 33

Inserisci prodotto 34

Use case 34

Use case diagram 34

Object Entity, Boundary, Control 40

Modifica prezzo 40

Use case 40

Use case diagram 41

Mockup 41

Object Entity, Boundary, Control 48

Use case clienti 49

Acquisto prodotto 49

Use case 49

Use case diagram 49

Mockup 49

Object Entity, Boundary, Control 56

Richiesta preventivo 57

Use case 57

Use case diagram 57

Mockup 57

Object Entity, Boundary, Control 61

Conferma preventivo 61

Use case 61

Use case diagram 62

Mockup 62

Object Entity, Boundary, Control 66

**Top Manager:**

*Prof. De Lucia Andrea*

**Team di sviluppo:**

|  |  |
| --- | --- |
| Nome e Cognome | Matricola |
| *Stefano Foresta* | 0512100504 |
| *Gennaro Franzese* | 0512100270 |

**Revision History:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Autore** | **Data** | **Descrizione** | **Versione** |
| *Stefano Foresta* | *19/10/16* | *Struttura documento* | *v 1.0* |
| *Gennaro Franzese* | *20/10/16* | *Stesura generale del documento* | *v 2.0* |
| *Stefano Foresta* | *01/11/16* | *Use Case Diagram* | *v 3.0* |
| *Stefano Foresta* | *02/11/16* | *Mockup utente generico* | *v 4.0* |
| *Gennaro Franzese* | *03/11/16* | *Inserimento Object boundary, Object identity, Control Object* | *v 5.0* |
| *Foresta Stefano* | *03/11/16* | *Inserimento Mockup cliente, Mockup amministratore* | *v 5.1* |
| *Foresta Stefano* | *04/11/16* | *Modifica di alcuni Mockup* | *v 5.2* |

# Introduzione

## Descrizione del sistema

Sistema informatizzato di gestione di un negozio di informatica. In primo luogo si prevede la realizzazione di un database che permetta la catalogazione digitale degli addetti ai lavori, dei fornitori, dei prodotti/servizi offerti e dei clienti. E in secondo luogo la realizzazione di una piattaforma web che permetta la gestione del negozio da parte dell’amministratore e la possibilità di acquisto da parte dei clienti.

## Scopo del sistema

Il sistema, automatizzando la maggior parte delle operazioni che si svolgono in un normale negozio di informatica, renderà più fluido e leggero il carico di lavoro di chi è addetto all’amministrazione del suddetto negozio. Le operazioni di carico/scarico merci, di acquisto/vendita prodotti e di gestione documentazione (fatture in entrata/uscita) saranno gestite dal sistema che si verrà a creare e quindi i dati verranno conservati in un database relazionale creato ad-hoc e la gestione sarà affidata ad una web application disegnata su misura per il cliente.

## Obiettivi

L’obiettivo è quello di fornire una piattaforma semplice ed intuitiva sia per chi è cliente del negozio sia per chi gestisce lo stesso.

Le macro funzionalità che si vogliono principalmente offrire sono:

* Cliente: registrazione, login e logout, acquisto;
* Amministratore: registrazione addetti ai lavori, login e logout, carico merci, inserimento servizi/merci;

## Overview

Il presente documento R.A.D. (Requirements Analysis Document) descrive il sistema in termini di requisiti funzionali e non funzionali e serve come base contrattuale con il cliente.

# Architettura del sistema corrente

L’architettura attualmente proposta non va a sostituire nessuna architettura presente perché il sistema è alla sua prima versione.

# Sistema proposto

## Identificazione attori

**Amministratore:** è colui che gestisce la piattaforma, inizialmente si occupa di ogni aspetto funzionale del sistema ma è previsto, in un secondo momento, lo sviluppo di un modulo che possa prevedere la divisione dei compiti tra vari attori che gestiscono separatamente i vari aspetti che sono presenti (es. vendita, gestione magazzino, spedizioni).

**Cliente:** è il generico cliente che si registra, accede e acquista beni e/o servizi offerti dalla piattaforma.

## Requisiti funzionali

In questa sezione vengono specificati quali funzioni il sistema deve fornire per soddisfare i bisogni dei clienti e del gestore del negozio.

**RF\_1 - Gestione Registrazione**

Questa funzionalità consente ai clienti di registrarsi ad I-Tech inserendo i propri dati.

**Attore:** Cliente

**RF 1.1 - Registrazione studente:** Questa funzionalità permette ad un cliente di registrarsi al sistema in modo da poter effettuare il login per eseguire acquisti di beni e servizi.

**RF\_2 - Gestione Autenticazione**

Questa funzionalità è comune per tutti gli attori (clienti e gestori) e deve essere in grado di gestire l’autenticazione al sistema degli stessi.

**RF 2.1 - Login:** Questa funzionalità permette di far effettuare l’accesso al sistema autenticandosi ed avendo a disposizione le varie funzionalità offerte dalla piattaforma.

**RF 2.2 - Logout:** Questa funzionalità permette la disconnessione dal sistema.

**RF\_3 - Gestione Utenti**

Questa funzionalità permette la gestione degli utenti registrati alla piattaforma.

**Attore:** Amministratore

**RF\_3.1.1 - Comunicazione con i clienti:** questa funzionalità permette all’amministratore di comunicare le offerte presenti sulla piattaforma a tutti gli utenti registrati o di comunicare direttamente con gli utenti per gestire problematiche/ordini singolarmente.

**RF\_3.1.2 - Inserimento gestori piattaforma:** questa funzionalità permette all’amministratore di inserire nuovi utenti che devono gestire la pagina (magazzinieri, web manager, ecc.).

**RF\_3.1.3 - Modifica profilo:** questa funzionalità permette all’amministratore la modifica di alcuni parametri del profilo.

**RF\_3.1.4 - Visualizza profilo:** questa funzionalità permette all’amministratore la visualizzazione del profilo.

**Attore:** Cliente

**RF\_3.2.1 - Modifica profilo:** questa funzionalità permette al cliente la modifica di alcuni parametri del profilo.

**RF\_3.2.2 - Visualizza profilo:** questa funzionalità permette al cliente la visualizzazione del profilo.

**RF\_3.2.3 - Visualizza cronologia acquisti:** questa funzionalità permette al cliente la visualizzazione della cronologia dei suoi acquisti.

**RF\_4 - Gestione Ordine**

Questa funzionalità permette la gestione degli acquisti.

**Attore:** Amministratore

**RF\_4.1.1 – Presa in carico:** questa funzionalità permette all’amministratore di finalizzare la vendita ovvero preparare i prodotti per la spedizione.

**Attore:** Cliente

**RF\_4.2.1 - Visualizza prodotti:** questa funzionalità permette al cliente di navigare nel sito visualizzando i prodotti in vendita.

**RF\_4.2.2 - Carrello:** questa funzionalità permette al cliente di navigare nel sito inserendo nel carrello i prodotti che vuole acquistare.

**RF\_4.2.3 - Acquisto:** questa funzionalità permette al cliente di finalizzare l’acquisto dei prodotti scelti.

**RF\_5 - Gestione Magazzino**

Questa funzionalità permette la gestione del magazzino.

**Attore:** Amministratore

**RF\_5.1 - Inserimento prodotto:** questa funzionalità permette al magazziniere di aggiornare la lista dei prodotti presenti nel negozio;

## Requisiti non funzionali

**RNF\_1 - Usabilità**

Il sistema deve rispettare il principio di usabilità che può essere articolato come segue:

**- Sistema Efficace:** il sistema deve fornire strumenti precisi e completi con cui l’utente può raggiungere velocemente un obiettivo specifico.

**- Sistema Efficiente:** attraverso l’utilizzo di risorse competitive e complete che permettono agli utenti maggiore velocità nell’utilizzo della piattaforma.

**- Sistema Soddisfacente:** il sistema deve rilasciare feedback continui che permettono all’utente di orientarsi al meglio e di poter avere meno problemi possibili durante l’utilizzo della piattaforma.

**RNF\_2 - Affidabilità**

Il sistema è basato sull’architettura Client-Server, entrambi i componenti devono essere affidabili quindi devono essere in grado di poter mantenere i propri dati anche in caso di guasti (problemi di fornitura elettrica, usura dell'hardware del server, attacchi informatici, problemi legati al browser, interruzione della connettività alla rete ecc.).

Il sistema dev’essere inoltre usufruibile 24 ore su 24.

Il sistema deve prevedere periodicamente la possibilità di effettuare dei backup.

**RNF\_3 - Performance**

Il sistema, inizialmente, deve essere in grado di gestire 100 connessioni contemporanee.

Deve quindi poter gestire insieme vendite, caricamento merce e mail service senza che gli utilizzatori del sistema si accorgano del carico di lavoro che il server gestisce.

**RNF\_4 - Manutenibilità**

Il sistema verrà implementato con un’architettura three-tier, così eventuali modifiche ad uno dei tre moduli non comporterà il blocco del sistema e la relativa modifica dei blocchi rimanenti rispetto a quello aggiornato.

**RNF\_5 - Implementazione**

Back-end del sistema realizzato in Php.

**RNF\_6 - Estensibilità**

La documentazione che si andrà a produrre e il modo in cui l’applicazione verrà implementata permetterà lo sviluppo e l’integrazione di moduli aggiuntivi.

# Modelli del sistema

Si prevedono inizialmente due tipi di attori: amministratore e cliente.

## Scenari amministratore

|  |  |
| --- | --- |
| **Nome scenario** | SC\_2.1\_Inserimento\_prodotti |
| **Partecipanti** | Giuseppe: Amministratore |
| **Flusso di eventi** | 1. Giuseppe riceve la merce dal fornitore; 2. Giuseppe controlla la merce; 3. Giuseppe decide quindi di accedere alla piattaforma; 4. Giuseppe compila il form di accesso inserendo username e password; 5. Giuseppe accede al form di inserimento merce; 6. Giuseppe inizia a inserire tutti i prodotti ricevuti dal fornitore nel database; 7. Completato l’inserimento Giuseppe esegue il logout. |

|  |  |
| --- | --- |
| **Nome scenario** | SC\_2.2\_Acquisizione\_vendita |
| **Partecipanti** | Umberto: Amministratore |
| **Flusso di eventi** | 1. Umberto compila il form di accesso inserendo username e password; 2. Umberto accede alla sezione vendite; 3. Umberto controlla la merce venduta; 4. Umberto conferma l’avvenuta acquisizione dell’ordine; 5. Il sistema comunica ai clienti l’accettazione dell’ordine. |

## Scenari cliente

|  |  |
| --- | --- |
| **Nome scenario** | SC\_1.1\_Registrazione\_utente |
| **Partecipanti** | Pino: cliente |
| **Flusso di eventi** | 1. Pino vuole acquistare un computer portatile ed accede per la prima volta ad I-Tech; 2. Naviga nel sito ed entra nella sezione dedicata ai computer portatili; 3. Scorre tra i vari modelli e trova il computer portatile adatto alle sue esigenze e decide di procedere con l’acquisto; 4. Lo inserisce nel carrello e procede all’acquisto; 5. Il sistema lo avverte che per procedere all’acquisto di un prodotto è necessario accedere alla piattaforma o registrarsi ex novo; 6. Pino decide quindi di registrarsi dato che è il suo primo accesso alla piattaforma; 7. Pino compila il form di registrazione inserendo tutti i dati obbligatori; 8. Pino conferma la registrazione. |

|  |  |
| --- | --- |
| **Nome scenario** | SC\_1.2\_Acquisto |
| **Partecipanti** | Pino: cliente |
| **Flusso di eventi** | 1. Pino vuole acquistare un computer portatile ed accede per la prima volta ad I-Tech; 2. Naviga nel sito ed entra nella sezione dedicata ai computer portatili; 3. Scorre tra i vari modelli e trova il computer portatile adatto alle sue esigenze e decide di procedere con l’acquisto; 4. Lo inserisce nel carrello e procede all’acquisto; 5. Il sistema lo avverte che per procedere all’acquisto di un prodotto è necessario accedere alla piattaforma o registrarsi ex novo; 6. Pino decide quindi di accedere alla piattaforma; 7. Pino compila il form di accesso inserendo username e password; 8. Pino conferma l’accesso; 9. Pino procede all’acquisto e la piattaforma lo reindirizza alla pagina di conferma dati; 10. Pino controlla l’indirizzo di spedizione e procede; 11. Il sistema chiede all’utente se vuole procedere con il pagamento attraverso il metodo predefinito o con un altro metodo; 12. Pino sceglie il metodo di pagamento e conferma l’acquisto; 13. Il sistema inoltra in automatico una mail con il riepilogo dell’acquisto appena effettuato. |

|  |  |
| --- | --- |
| **Nome scenario** | SC\_1.3\_Riparazione |
| **Partecipanti** | Pino: cliente |
| **Flusso di eventi** | 1. Pino vuole riparare uno smartphone; 2. Pino accede ad I-Tech; 3. Pino decide quindi di accedere alla piattaforma; 4. Pino compila il form di accesso inserendo username e password; 5. Pino conferma l’accesso; 6. Pino sceglie dalla sua area utente la voce “riparazione”; 7. Pino compila il form per la riparazione inserendo le specifiche; 8. Pino inoltra la domanda di riparazione; 9. Il sistema informa l’utente che la richiesta è stata presa in carico e che il preventivo di riparazione verrà inviato all’utente entro 24h lavorative; 10. (una volta ricevuta la risposta) Pino controlla il preventivo e decide di riparare il telefono; 11. Accetta quindi il preventivo e segue le istruzioni per la riparazione. |

# Casi d’uso

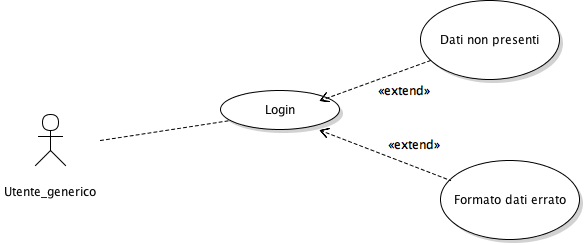
## Use case utente generico: amministratore, cliente

### Login

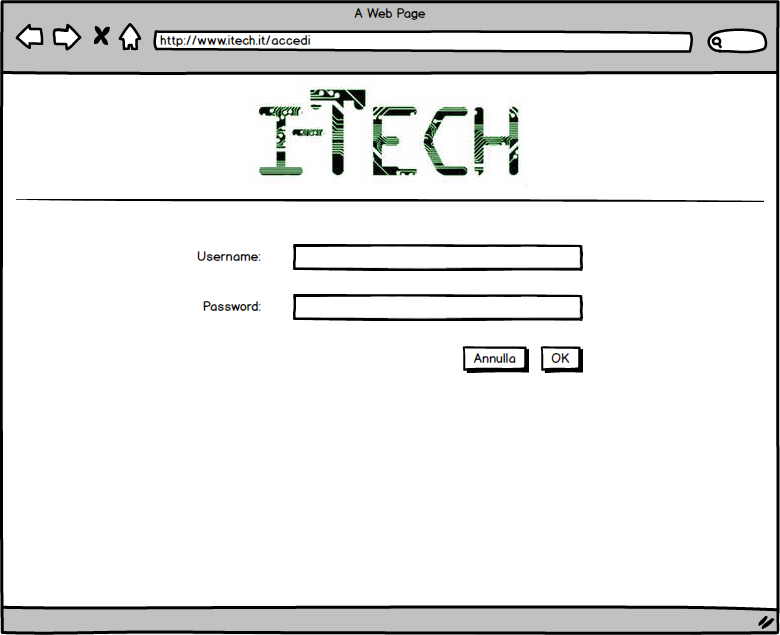
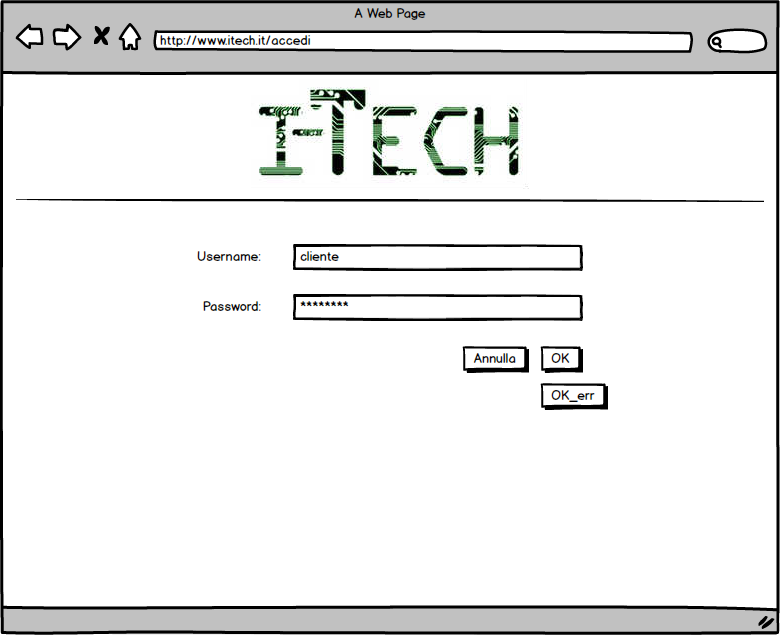
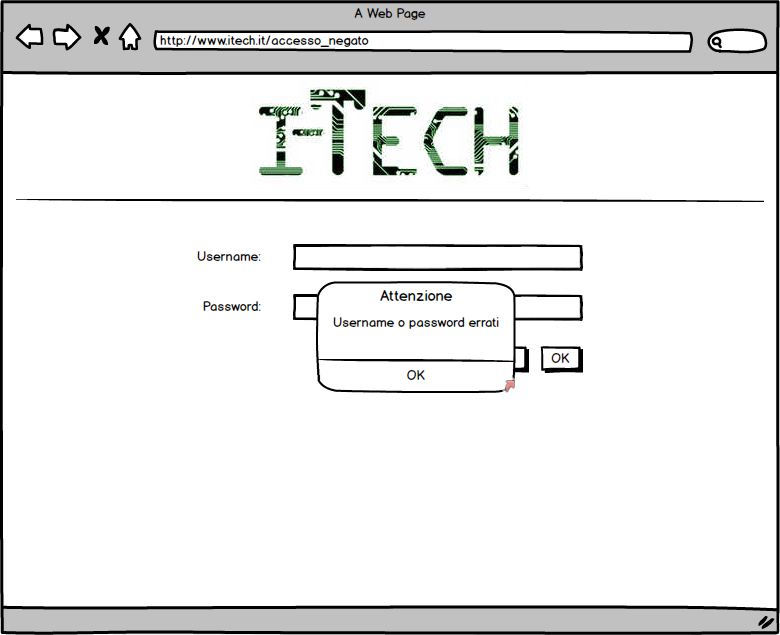
#### Use case

|  |  |
| --- | --- |
| **Nome use case** | Login |
| **Partecipanti** | Utente generico |
| **Condizioni di ingresso** | L’utente accede al sistema  L’utente è registrato al sistema |
| **Flusso di eventi** | 1. L’utente accede al sistema tramite motore di ricerca o tramite link diretto; 2. L’utente clicca sul pulsante accedi; 3. L’utente generico inserisce username e password e sottomette il form cliccando su login; 4. Il sistema riceve i dati e cerca l’username immesso e una volta trovato confronta la password inserita; 5. Viene notificato l’insuccesso dell’operazione o in caso di successo si viene reindirizzati nella pagina personale dell’utente che ha effettuato l’accesso. |
| **Eccezioni** | Dati non presenti  Errore nel formato dei dati |
| **Condizioni di uscita** | L’utente ha avuto accesso al sistema |

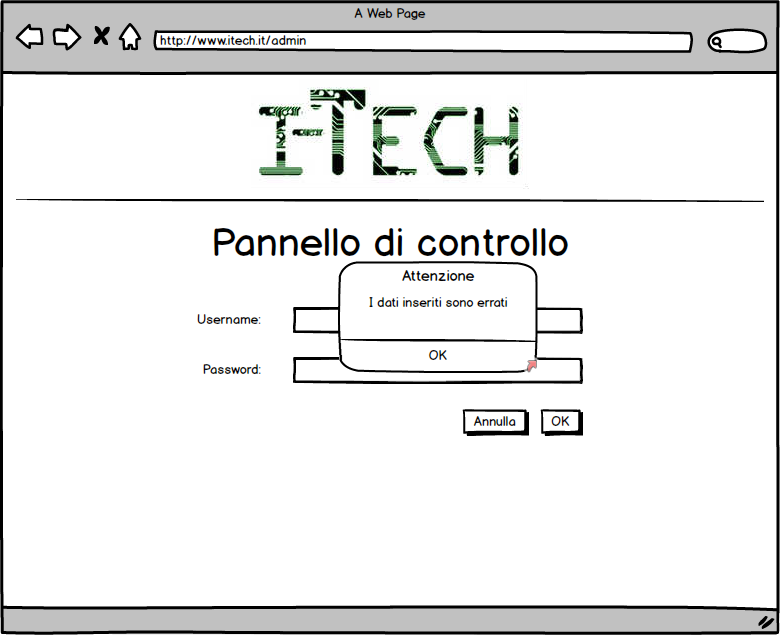
#### Use case diagram



#### Mockup - cliente

1. 
2. 
3. 
   1. Errore 
4. 

#### Mockup – Amministratore

1. 
2. 
   1. Errore 
3. 

#### Object Entity, Boundary, Control

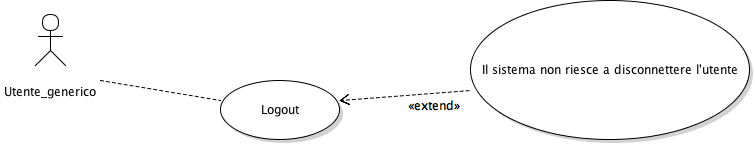
|  |  |
| --- | --- |
| **Object Entity** | |
| ***Oggetto*** | ***Descrizione*** |
| *Cliente* | Contiene gli attributi del cliente |
| *Amministratore* | Contiene gli attributi dell’amministratore |
| **Object Boundary** | |
| ***Oggetti*** | ***Descrizione*** |
| Button\_accedi | Porta alla pagina di inserimento dei dati per l’accesso |
| Login\_form | Form per l’inserimento dei dati |
| **Object Control** | |
| ***Oggetti*** | ***Descrizione*** |
| Login | Richiama la funzione Login che verifica la corrispondenza dei dati inseriti nel Login\_form |

### Logout

#### Use case

|  |  |
| --- | --- |
| **Nome use case** | Logout |
| **Partecipanti** | Utente generico |
| **Condizioni di ingresso** | L’utente ha effettuato l’accesso |
| **Flusso di eventi** | 1. L’utente clicca su logout; 2. Il sistema disconnette l’utente. |
| **Eccezioni** | Il sistema non riesce a disconnettere l’utente |
| **Condizioni di uscita** | L’utente ha effettuato il logout |

#### Use case diagram



#### Mockup – Cliente

1. 
2. 

#### Mockup – Amministratore

1. 
2. 
3. 

#### Object Entity, Boundary, Control

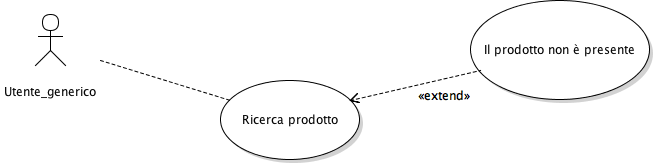
|  |  |
| --- | --- |
| **Object Entity** | |
| ***Oggetto*** | ***Descrizione*** |
| *Cliente* | Contiene gli attributi del cliente |
| *Amministratore* | Contiene gli attributi dell’amministratore |
| **Object Boundary** | |
| ***Oggetti*** | ***Descrizione*** |
| Button\_esci | Pulsante per il logout dell’utente |
| **Object Control** | |
| ***Oggetti*** | ***Descrizione*** |
| Logout | Chiude la sessione dell’utente |

### Ricerca prodotto

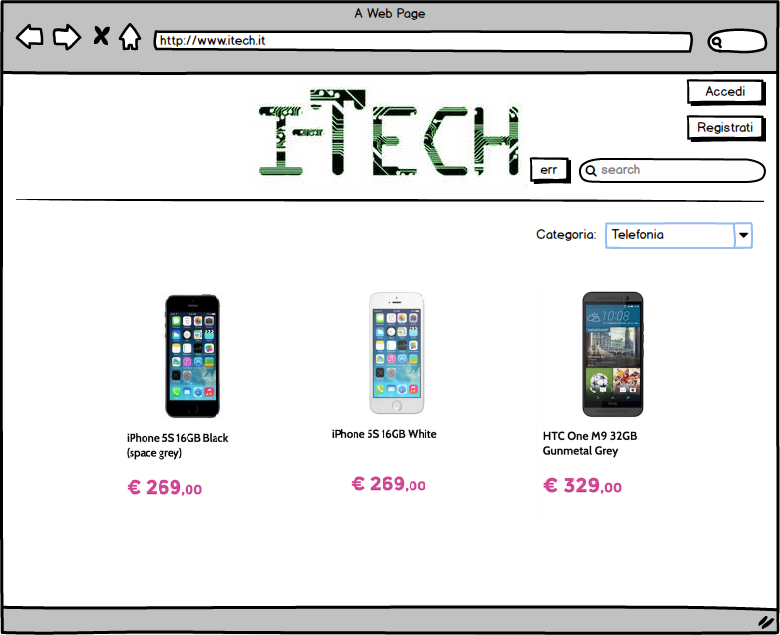
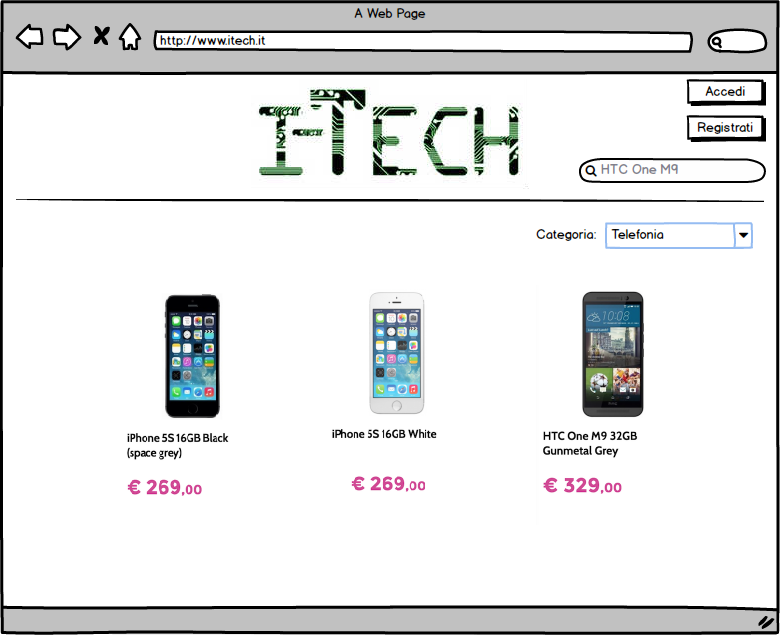
#### Use case

|  |  |
| --- | --- |
| **Nome use case** | Ricerca prodotto |
| **Partecipanti** | Utente generico |
| **Condizioni di ingresso** | L’utente sta navigando il sito web |
| **Flusso di eventi** | 1. L’utente naviga sul sito web; 2. Decide di cercare un prodotto specifico; 3. Digita sulla barra di ricerca il nome del prodotto; 4. L’utente lancia la ricerca; 5. Il sistema visualizza i prodotti trovati. |
| **Eccezioni** | Il sistema non trova i prodotti desiderati |
| **Condizioni di uscita** | L’utente trova il prodotto |

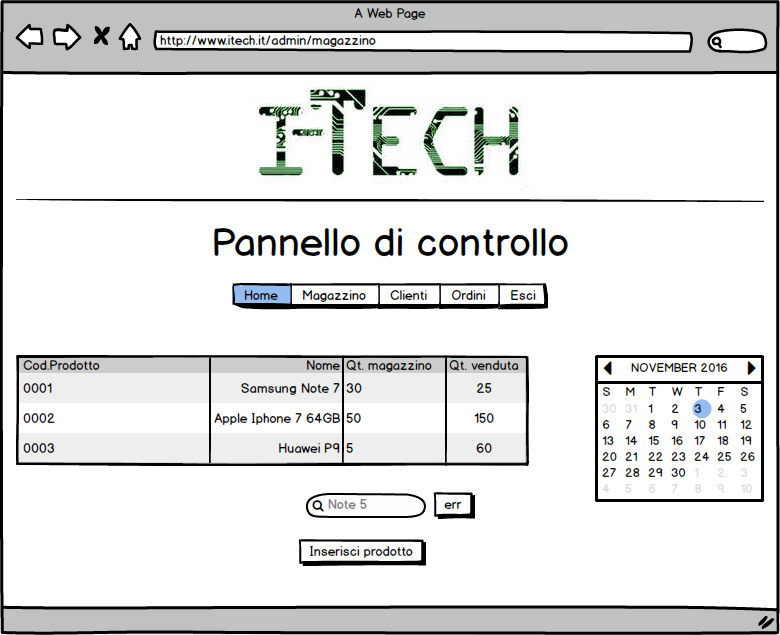
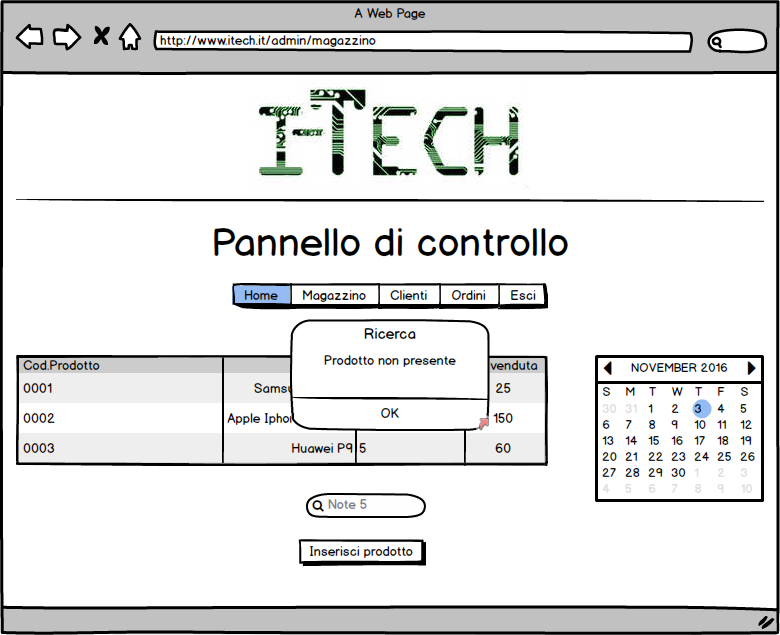
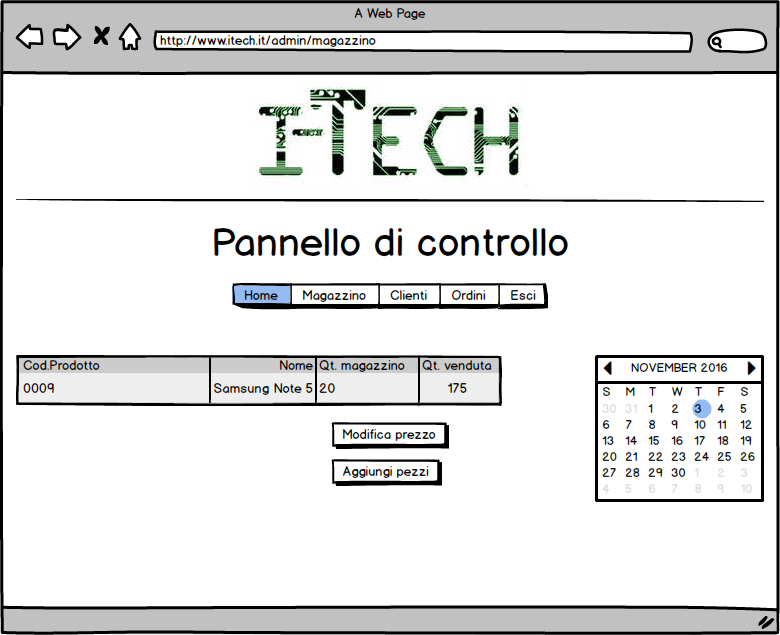
#### Use case diagram



#### Mockup - cliente

1. 
2. 
   1. 
3. 

#### Mockup – amministratore

1. 
2. 
   1. Errore 
3. 

#### Object Entity, Boundary, Control

|  |  |
| --- | --- |
| **Object Entity** | |
| ***Oggetto*** | ***Descrizione*** |
| *Cliente* | Contiene gli attributi del cliente |
| *Amministratore* | Contiene gli attributi dell’amministratore |
| *Prodotto* | Contiene gli attributi dei prodotti |
| **Object Boundary** | |
| ***Oggetti*** | ***Descrizione*** |
| Form\_ricerca | Form per l’inserimento del nome del prodotto da cercare |
| Button\_cerca | Passa i parametri inseriti alla funzione di ricerca |
| **Object Control** | |
| ***Oggetti*** | ***Descrizione*** |
| Ricerca | Richiama la funzione ‘cerca’ che restituisce i prodotti trovati nel database |

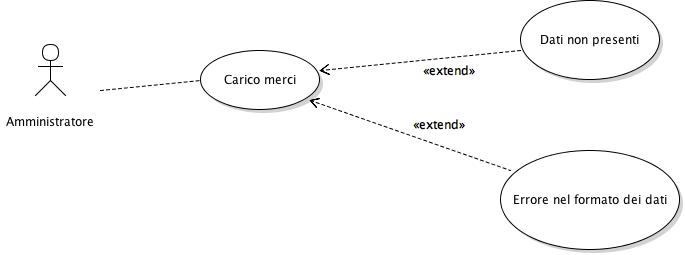
## Use case amministratore

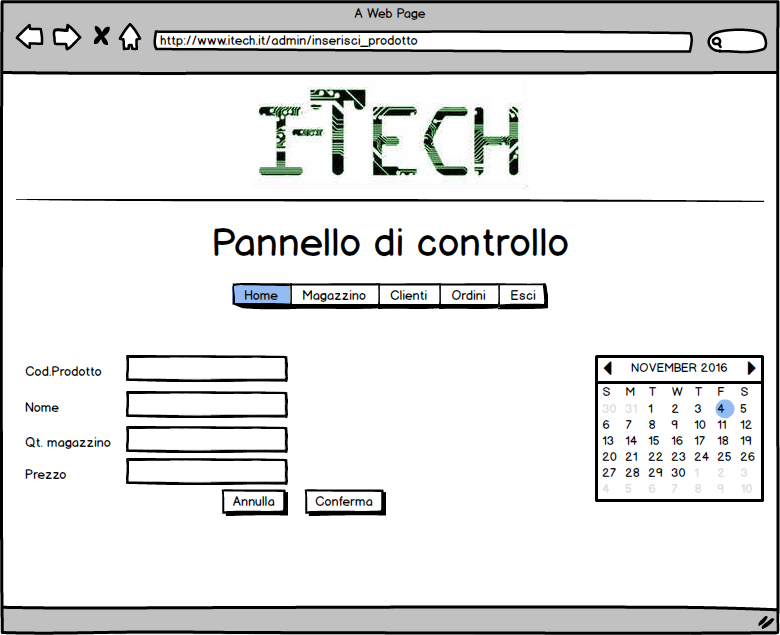
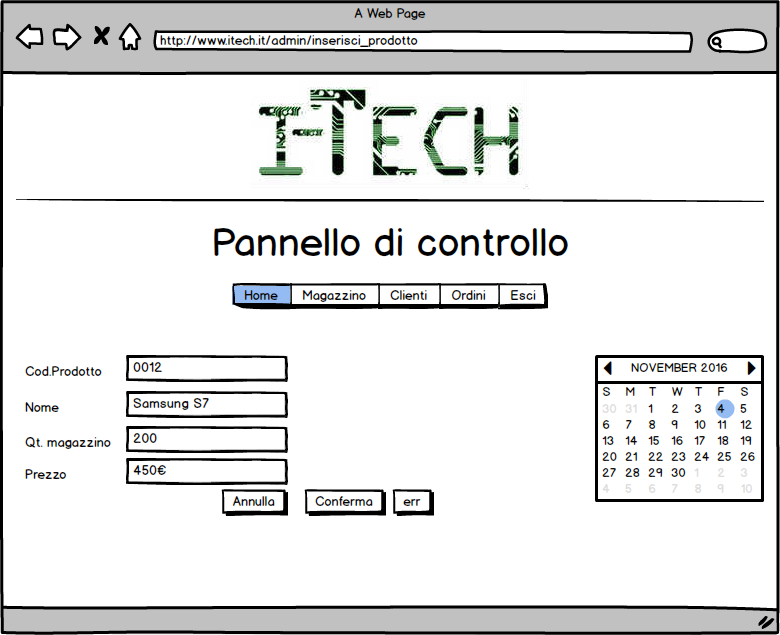
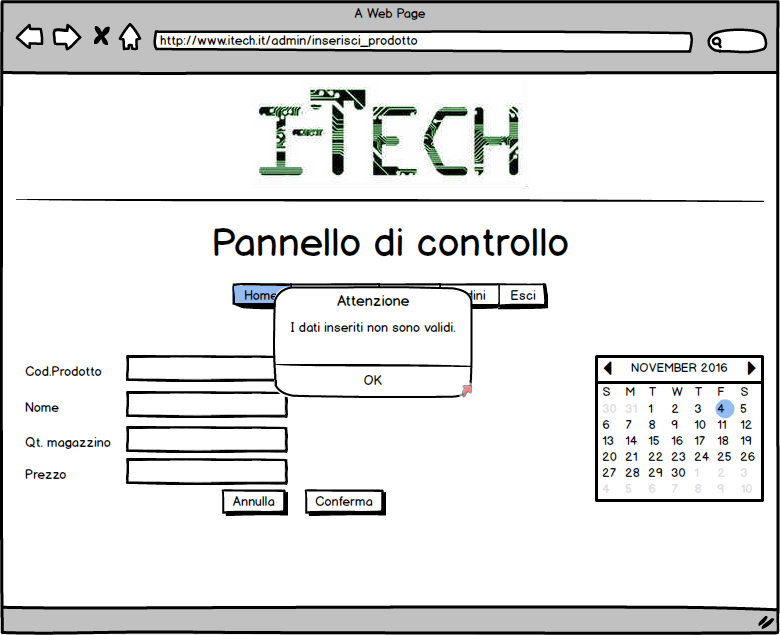
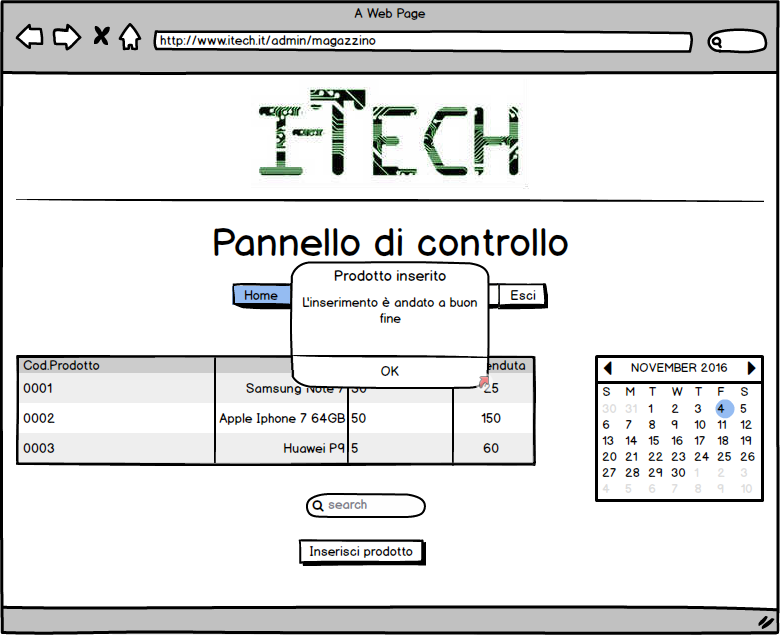
### Inserisci prodotto

#### Use case

|  |  |
| --- | --- |
| **Nome use case** | Inserisci prodotto |
| **Partecipanti** | Amministratore |
| **Condizioni di ingresso** | L’amministratore ha effettuato il login |
| **Flusso di eventi** | 1. L’amministratore si reca sulla sezione “Magazzino”; 2. Clicca su “Inserisci prodotti” 3. Compila il form con i prodotti da inserire; 4. Conferma l’inserimento; 5. Il sistema resetta i campi e permette l’immissione di un nuovo prodotto; 6. L’amministratore ripete i passi 2 e 3 per tutti i prodotti da inserire; |
| **Eccezioni** | Dati non presenti  Errore nel formato dei dati |
| **Condizioni di uscita** | Sono stati inseriti tutti i prodotti |

#### Use case diagram



1. 
2. 
3. 
   1. 
4. 
5. 

#### Object Entity, Boundary, Control

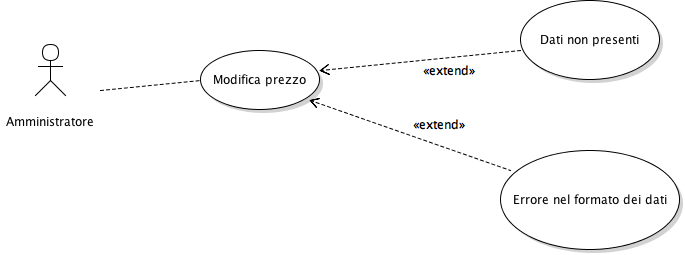
|  |  |
| --- | --- |
| **Object Entity** | |
| ***Oggetto*** | ***Descrizione*** |
| *Amministratore* | Contiene gli attributi dell’amministratore |
| *Prodotto* | Contiene gli attributi dei prodotti |
| **Object Boundary** | |
| ***Oggetti*** | ***Descrizione*** |
| Form\_prodotto | Form per l’inserimento del prodotto e dei suoi attributi |
| Button\_conferma | Passa i parametri inseriti alla funzione di inserimento |
| **Object Control** | |
| ***Oggetti*** | ***Descrizione*** |
| Inserimento | Richiama la funzione ‘inserisci’ che aggiunge il prodotto nella tabella relativa |

### Modifica prezzo

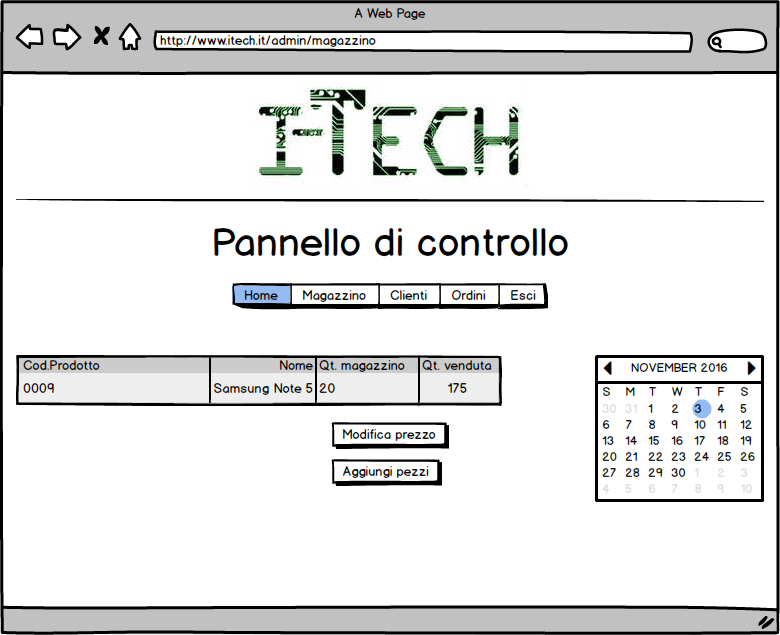
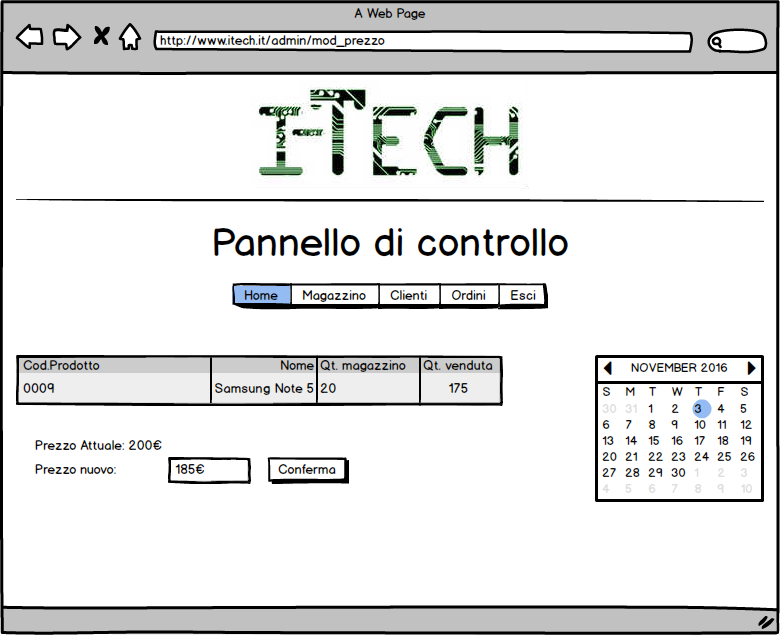
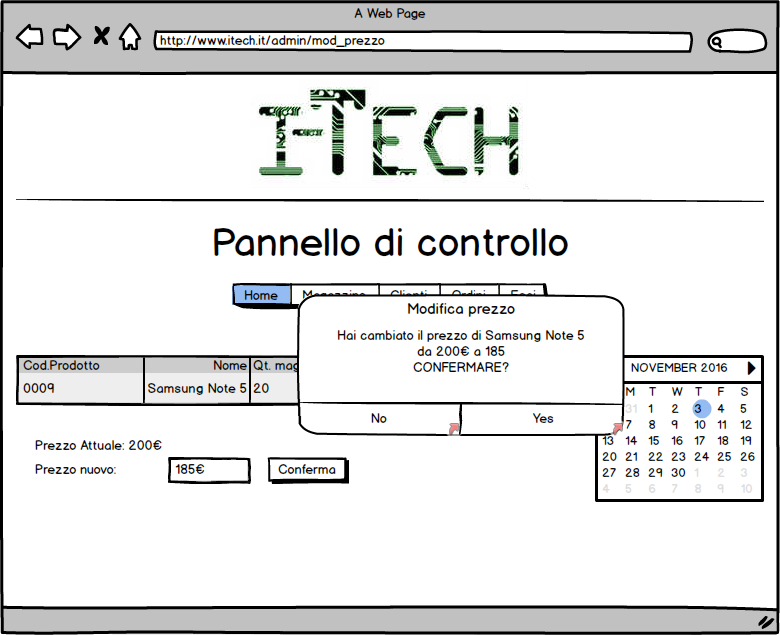
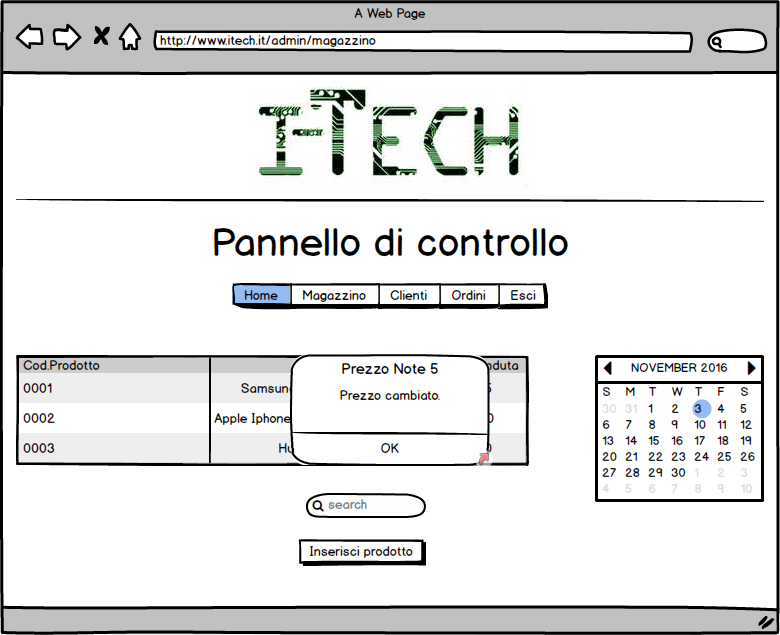
#### Use case

|  |  |
| --- | --- |
| **Nome use case** | Modifica prezzo |
| **Partecipanti** | Amministratore |
| **Condizioni di ingresso** | L’amministratore ha effettuato il login  L’amministratore ha ricercato il prodotto |
| **Flusso di eventi** | 1. L’amministratore clicca su “modifica prezzo”; 2. L’amministratore compila il form con il nuovo prezzo; 3. L’amministratore conferma; 4. Il sistema modifica il prezzo. |
| **Eccezioni** | Dati non presenti  Errore nel formato dei dati |
| **Condizioni di uscita** | Il sistema ha modificato il prezzo |

#### Use case diagram



#### Mockup

1. 
2. 
3. 
4. 
   1. NO 
   2. SI 
5. 

#### Object Entity, Boundary, Control

|  |  |
| --- | --- |
| **Object Entity** | |
| ***Oggetto*** | ***Descrizione*** |
| *Amministratore* | Contiene gli attributi dell’amministratore |
| *Prodotto* | Contiene gli attributi dei prodotti |
| **Object Boundary** | |
| ***Oggetti*** | ***Descrizione*** |
| Form\_ricerca | Form per l’inserimento del nome del prodotto da cercare |
| Button\_cerca | Passa i parametri inseriti alla funzione di ricerca |
| Form\_modifica\_prezzo | Form per l’inserimento del prezzo del prodotto |
| Button\_modifica | Passa i parametri inseriti alla funzione di modifica prezzo |
| **Object Control** | |
| ***Oggetti*** | ***Descrizione*** |
| Ricerca | Richiama la funzione ‘cerca’ che restituisce i prodotti trovati nel database |
| Modifica\_prezzo | Richiama la funzione che modifica il prezzo del prodotto precedentemente cercato |

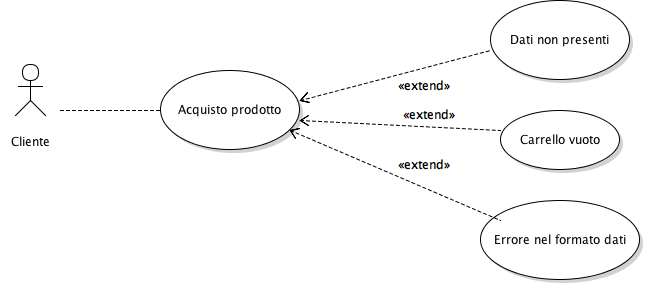
## Use case clienti

### Acquisto prodotto

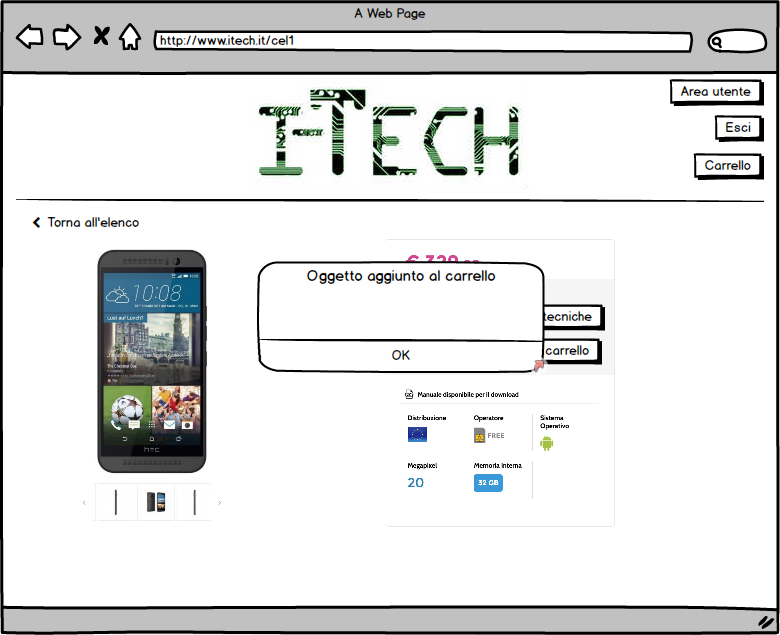
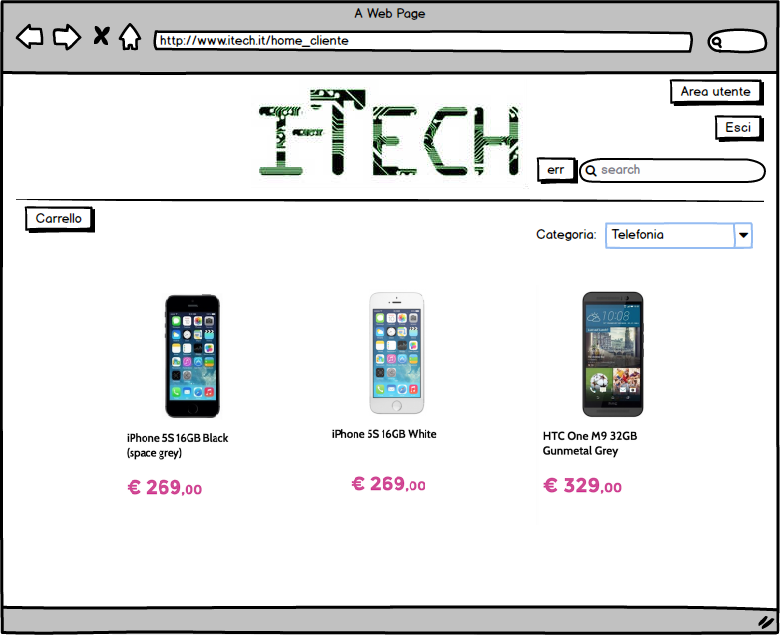
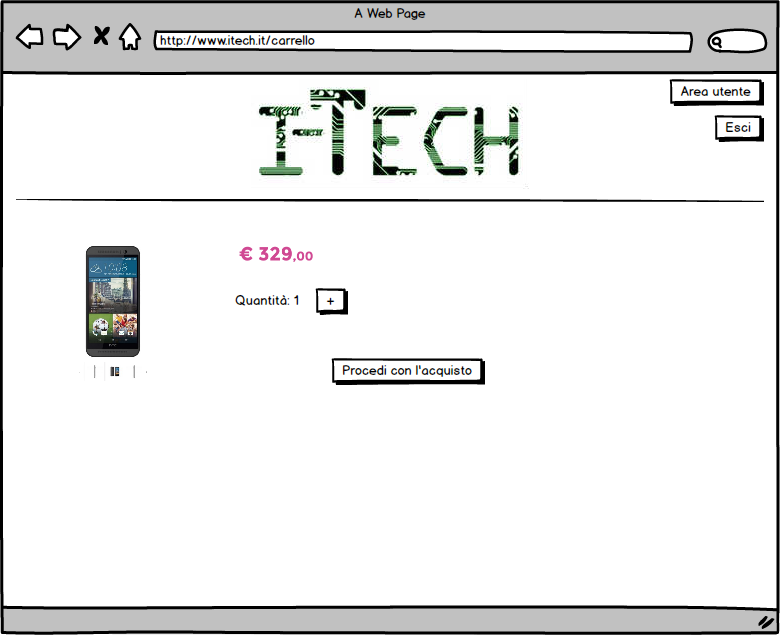
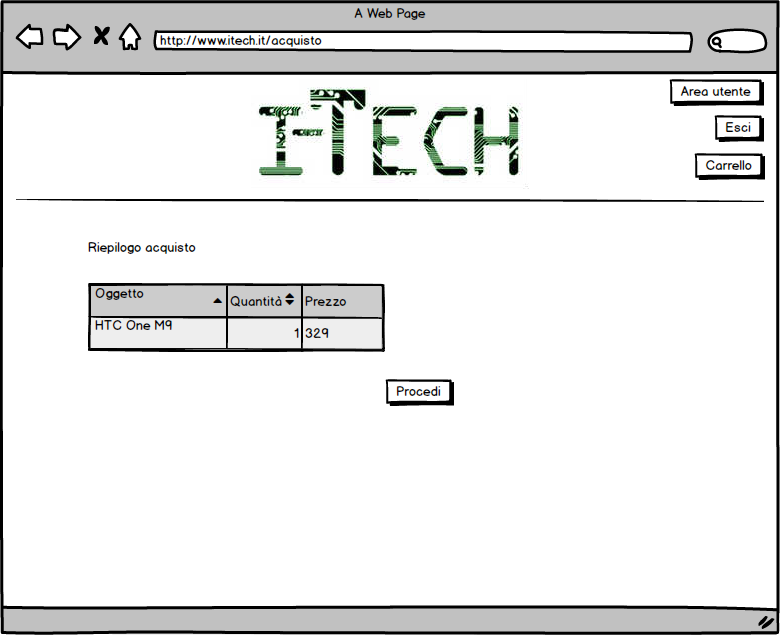
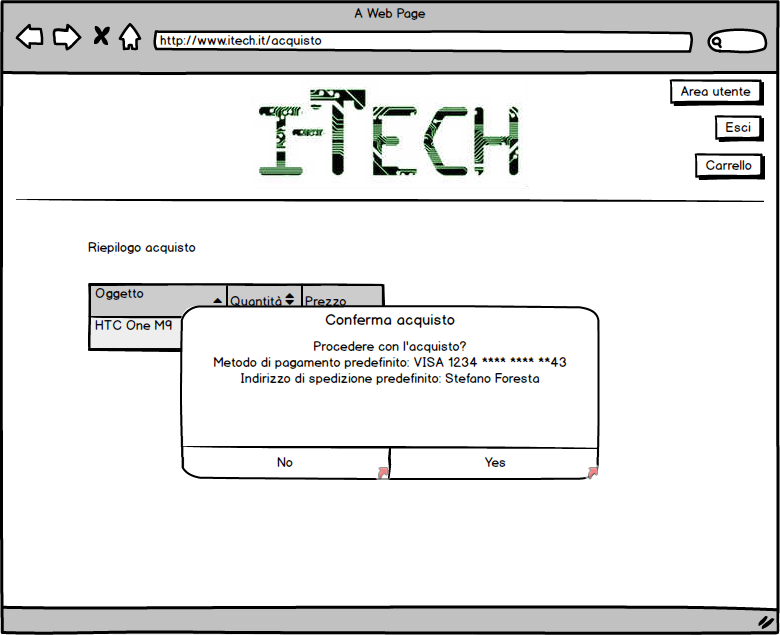
#### Use case

|  |  |
| --- | --- |
| **Nome use case** | Acquisto prodotto |
| **Partecipanti** | Cliente |
| **Condizioni di ingresso** | Il cliente ha effettuato il login  Il cliente ha ricercato il prodotto |
| **Flusso di eventi** | 1. Il cliente aggiunge il prodotto al carrello; 2. Il cliente clicca su “procedi con l’acquisto”; 3. Il cliente inserisce i dati per la spedizione; 4. Il cliente inserisce il metodo di pagamento; 5. Il cliente clicca su prosegui; 6. Il sistema riassume i dati relativi all’acquisto; 7. Il cliente conferma l’acquisto. |
| **Eccezioni** | Dati non presenti  Carrello vuoto  Errore nel formato dei dati |
| **Condizioni di uscita** | L’acquisto è andato a buon fine |

#### Use case diagram



#### Mockup

1. 
2. 
3. 
4. 
5. 
6. 
7. 

#### Object Entity, Boundary, Control

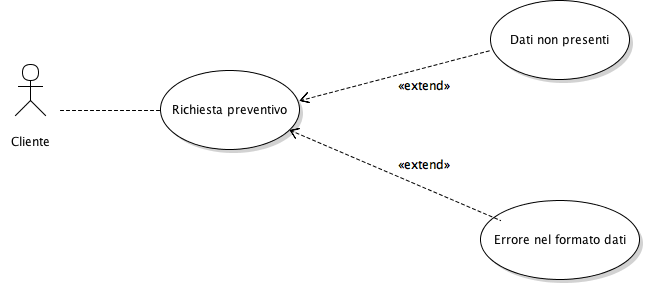
|  |  |
| --- | --- |
| **Object Entity** | |
| ***Oggetto*** | ***Descrizione*** |
| *Cliente* | Contiene gli attributi del cliente |
| *Prodotto* | Contiene gli attributi dei prodotti |
| *Ordine* | Contiene gli attributi dell’ordine |
| **Object Boundary** | |
| ***Oggetti*** | ***Descrizione*** |
| Form\_prodotto | Form che contiene il prodotto selezionato |
| Button\_aggiungi\_al\_carrello | Inserisce il prodotto nel carrello virtuale del cliente |
| Form\_carrello | Form contenente il riepilogo dei prodotti selezionati per l’acquisto |
| Button\_conferma | Passa codice prodotto e quantità alla funzione ‘ordine’ |
| Form\_pagamento | Form che contiene i dettagli di pagamento |
| **Object Control** | |
| ***Oggetti*** | ***Descrizione*** |
| Ordine\_prodotto | Elimina dal database la quantità di prodotti che sono stati acquistati |
| Finalizza\_acquisto | Invia i dettagli dell’ordine all’amministratore e conferma il successo dell’acquisto al cliente e il relativo pagamento |

### Richiesta preventivo

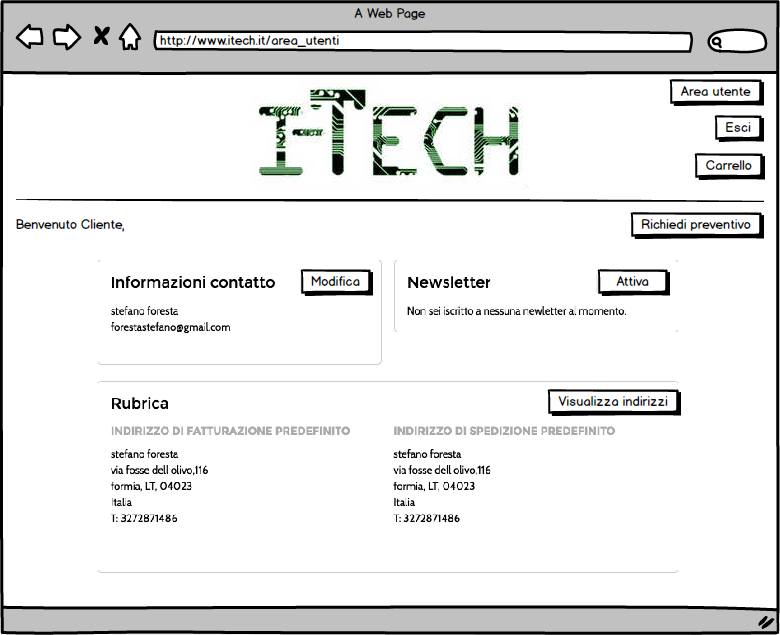
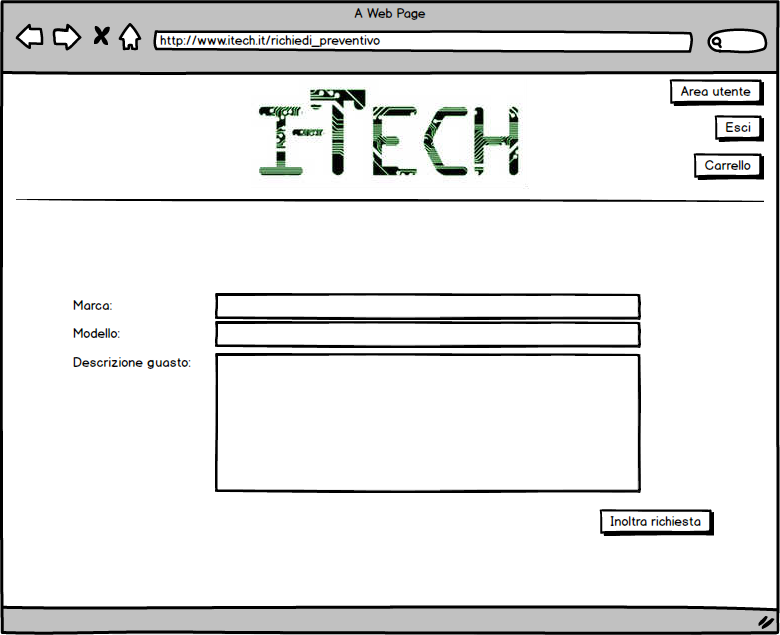
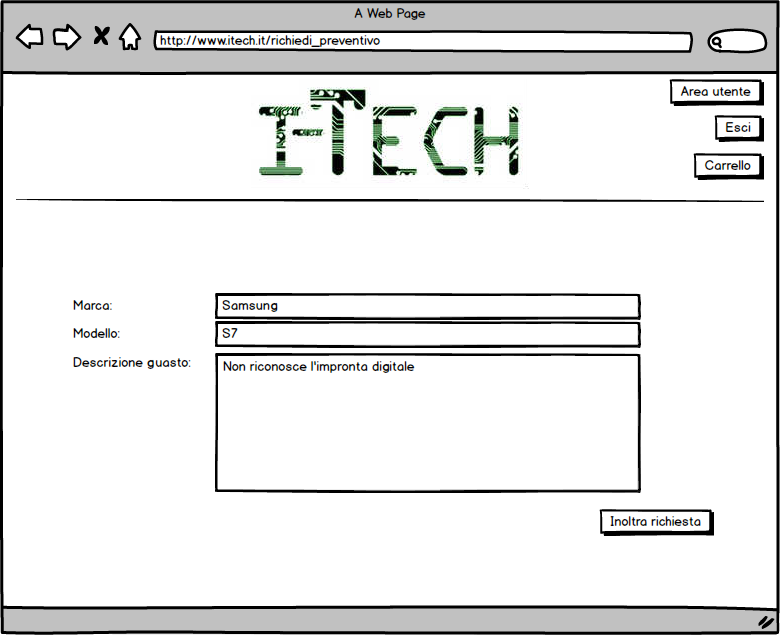
#### Use case

|  |  |
| --- | --- |
| **Nome use case** | Richiesta preventivo |
| **Partecipanti** | Cliente |
| **Condizioni di ingresso** | Il cliente ha effettuato il login |
| **Flusso di eventi** | 1. Il cliente clicca su “Richiedi preventivo”; 2. Il sistema reindirizza il cliente sul form “preventivo”; 3. Il cliente compila il form; 4. Il cliente clicca su “Prosegui con il preventivo”; 5. Il cliente visualizza il riepilogo del preventivo; 6. Il cliente controlla il preventivo e clicca su “Conferma preventivo”; 7. Il sistema elabora la richiesta e la indirizza all’amministratore. |
| **Eccezioni** | Dati non presenti  Errore nel formato dei dati |
| **Condizioni di uscita** | Il preventivo è stato inviato |

#### Use case diagram



#### Mockup

1. 
2. 
3. 
4. 

#### Object Entity, Boundary, Control

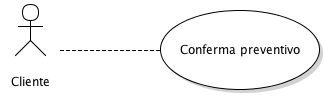
|  |  |
| --- | --- |
| **Object Entity** | |
| ***Oggetto*** | ***Descrizione*** |
| *Cliente* | Contiene gli attributi del cliente |
| *Preventivo* | Contiene gli attributi del preventivo |
| **Object Boundary** | |
| ***Oggetti*** | ***Descrizione*** |
| *Form\_preventivo* | Form che contiene le text per la richiesta del preventivo |
| *Button\_invia* | Inoltra la richiesta |
| **Object Control** | |
| ***Oggetti*** | ***Descrizione*** |
| Inoltro\_preventivo | Crea una richiesta di assistenza e la inoltra all’amministratore |

### Conferma preventivo

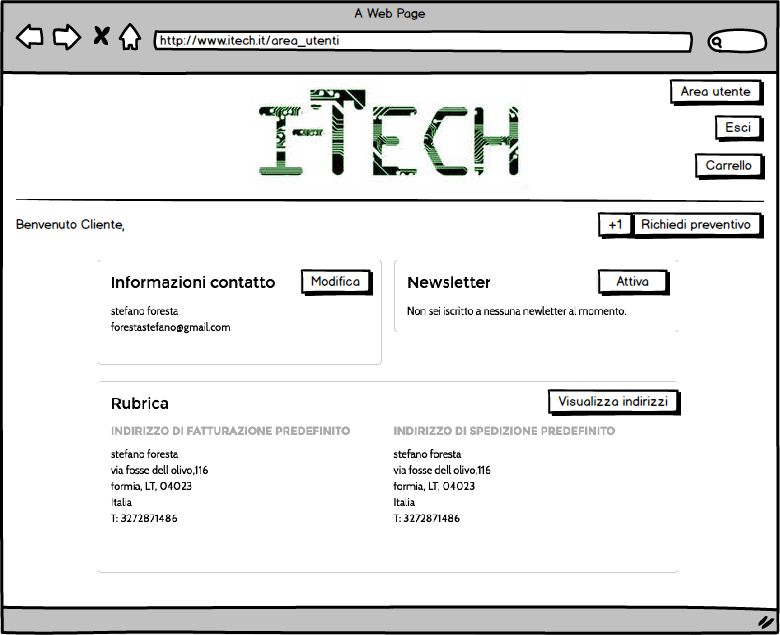
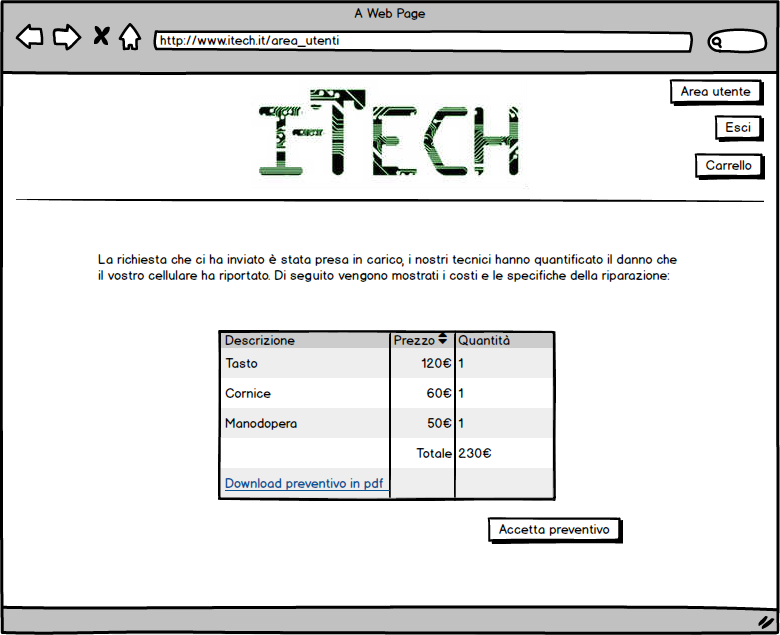
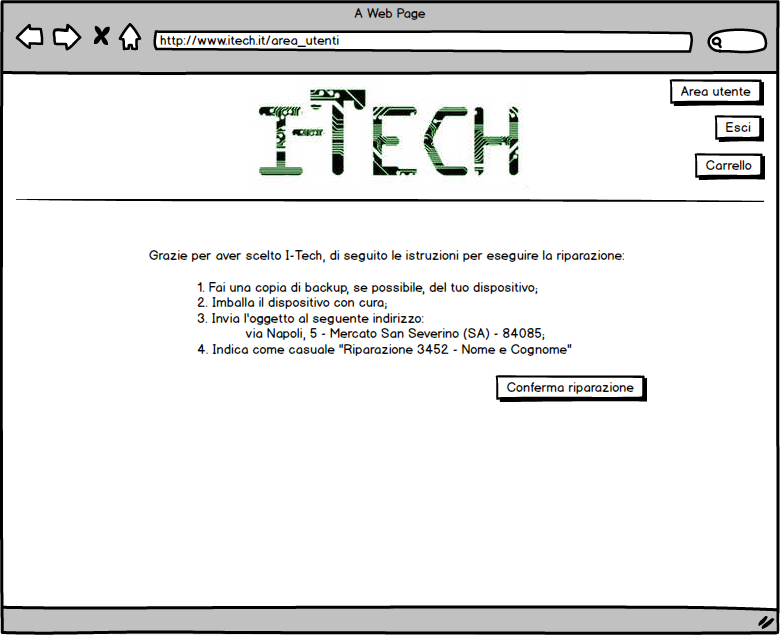
#### Use case

|  |  |
| --- | --- |
| **Nome use case** | Conferma preventivo |
| **Partecipanti** | Cliente |
| **Condizioni di ingresso** | Il cliente ha effettuato il login  Il cliente ha richiesto il preventivo  Il cliente ha ricevuto l’accettazione |
| **Flusso di eventi** | 1. Il cliente visualizza il prezzo del preventivo; 2. Il cliente accetta il preventivo; 3. Il sistema comunica l’accettazione all’amministratore; 4. Il cliente procede seguendo le istruzioni per la consegna dell’oggetto da riparare. |
| **Eccezioni** |  |
| **Condizioni di uscita** | Il cliente accetta il preventivo |

#### Use case diagram



#### Mockup

1. 
2. 
3. 
4. 

#### Object Entity, Boundary, Control

|  |  |
| --- | --- |
| **Object Entity** | |
| ***Oggetto*** | ***Descrizione*** |
| *Cliente* | Contiene gli attributi del cliente |
| *Preventivo* | Contiene gli attributi del preventivo |
| **Object Boundary** | |
| ***Oggetti*** | ***Descrizione*** |
| *Form\_preventivo* | Form che contiene il riepilogo del preventivo richiesto con i dettagli |
| *Button\_annulla* | Annulla il preventivo |
| *Button\_conferma* | Conferma il preventivo |
| **Object Control** | |
| ***Oggetti*** | ***Descrizione*** |
| Inoltr\_conferma\_preventivo | Accetta il preventivo e comunica all’amministratore che il cliente spedirà il prodotto seguendo le istruzioni inviate |