

3.7	operations searchForMembers, searchForMembersGraphic	15
3.8	operations getMemberlist, getMemberlistGraphic	15
3.9	operation sendRcd	15
4	Non-functional requirements	16
4.1	User interface	16
4.2	Browser compatibility	16
4.3	Scalability	16
4.4	Persistence	17
4.5	Session timeout	17
4.6	Integration	17
4.7	Programming style	17
5	Rules for development	18
5.1	What is allowed	18
5.2	What is not allowed	18
5.3	User feedback: The blog	18
5.3.1	How to post a message in the blog	19
5.4	Talking to the customer	19
5.5	Delivery	19

1 Introduction

1.1 Theme/purpose of the PbT system

PbT (People by Temperament) is a simple community portal where members can find others with whom they might like to get in contact: people register to become members, take a personality test, and then search for others based on criteria such as personality types, likes/dislikes etc. Members can then get in contact with one another if both choose to do so.

The system has both an interactive user interface via HTML pages and a WSDL/SOAP-based programmatic interface.

1.2 Requirements priorities and notation

The requirements described in this document are categorized into three different priority levels and each requirement is marked accordingly:

- | | |
|--------|---|
| MUST | • MUST marks an essential requirement. Unless all of these requirements are implemented, the system is considered unacceptable. |
| SHOULD | • SHOULD marks an important requirement. If some of these requirements are not implemented, the system is considered incomplete, but acceptable. |
| MAY | • MAY marks an optional requirement. These requirements are considered nice-to-have but need not be implemented when time is short or if their cost-benefit ratio is considered too high. |

Each requirement is marked by exactly one of these three terms, followed by a subscript number that is the unique reference number of that requirement (also repeated in the margin).

If a compound requirement (such as a use case) is marked MUST, that means at least the MUST requirements contained in it must be realized. It does not mean that all of its subrequirements must be realized.

If a compound requirement is marked SHOULD or MAY, it will be considered realized only if all of the MUST requirements contained in it are realized.

If you make tradeoffs among non-essential requirements, maximize the user value of the portal, rather than showing off your technical capabilities.

Phrases *in italics* in the use cases below are references to further use cases, the corresponding section number is appended in parentheses as a hyperlink.